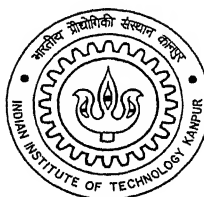


***APPLICATION OF STEREOLOGY AND IMAGE
ANALYSIS IN THE CHARACTERIZATION OF
TRIPLE JUNCTIONS IN POLYCRYSTALS***

By

LIEUTENANT SANJAY CHHABRA



**DEPARTMENT OF MATERIALS AND METALLURGICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY, KANPUR**

JANUARY, 2002

19
19/11/2012/MME
पुणे
भारतीय प्रोचोदना
उत्पादित क्र० A137954

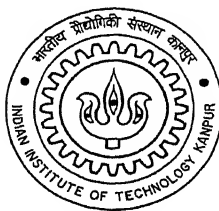


***APPLICATION OF STEREOLOGY AND IMAGE ANALYSIS IN THE
CHARACTERIZATION OF TRIPLE JUNCTIONS IN
POLYCRYSTALS***

***A THESIS SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS
FOR THE DEGREE OF***

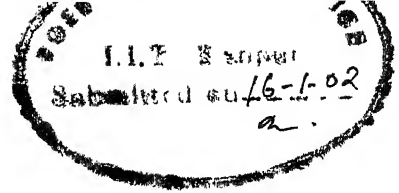
MASTER OF TECHNOLOGY

**BY
LIEUTENANT SANJAY CHHABRA**



**TO THE
DEPARTMENT OF MATERIALS AND METALLURGICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY, KANPUR
JANUARY 2002**

CERTIFICATE



**THIS IS TO CERTIFY THAT THE WORK CONTAINED IN THIS THESIS
TITLED**

***"APPLICATION OF STEREOLOGY AND IMAGE ANALYSIS IN
THE CHARACTERIZATION OF TRIPLE JUNCTIONS IN
POLYCRYSTALS"***

**BY LT SANJAY CHHABRA HAS BEEN CARRIED OUT UNDER MY
SUPERVISION AND THAT HIS WORK HAS NOT BEEN SUBMITTED
ELSEWHERE FOR ANY DEGREE**

A handwritten signature in black ink, appearing to read "Sandeep Sangal".

**(DR SANDEEP SANGAL)
PROFESSOR**

**DEPARTMENT OF MATERIALS AND METALLURGICAL ENGG
INDIAN INSTITUTE OF TECHNOLOGY
KANPUR - 208016**

JANUARY 2002

DEDICATED TO

MY PARENTS

**COL OM PRAKASH CHHABRA
AND
MRS USHA CHHABRA
(FOR THEIR BLESSINGS)**

**MY WIFE AND SON
MRS KIRAN
AND
KARAN
(FOR THEIR AFFECTION)**

Acknowledgments

I take this opportunity to express my deep indebtedness to Dr. Sandeep Sangal for his valuable guidance and supervision towards the completion of my M.Tech thesis work. His incessant encouragement and suggestions were invaluable. It was indeed an honor to work under him.

It was a great pleasure to be with Dr MN Mungole and I would like to thank him for advices and allowing me to use his laboratory facilities. I am also thankful to my counterparts Mr. John, Mr. Ramaiha and Mr. Chhillar for their generous support. My special thanks goes to Mr. G.J. Sunil who helped right from the beginning of my endeavors in this course.

Finally I most gratefully acknowledge my parents for their continuous support in each step of my life, without which it would have been impossible for me to reach this stage.

I am immeasurably grateful to my wife, Kiran, for her proper care and love throughout the busy period of this study. I also acknowledge the indirect and most important contribution of my little son, Karan, for refreshing the environment when needed most.

ABSTRACT

The dihedral angle distribution is a useful parameter in the quantification of polycrystalline microstructures. However, this microstructural parameter has received not much attention because the methodologies associated with the accurate measurements of dihedral angles are generally tedious and time consuming. In this study an automated methodology based on image processing and analysis, was developed to evaluate dihedral angles in microstructures in a two-dimensional plane section of polycrystals. It is shown that the technique is extremely fast and accurate and therefore can be applied to obtain a significantly large statistical sample of dihedral angles.

The study was expanded further to transform the plane dihedral angle distribution to true dihedral angle distributions using a 3-dimensional probe termed as “disector”. The true dihedral angles were further related to grain boundary energies to characterize polycrystals. This probe introduces precision as well as efficiency over the classical single as well as serial sectioning technique. This technique gives more information than a single 2-dimensional probe and is far less tedious and less complicated than the serial sectioning technique.

The application of the above methods for calculating plane and the true dihedral angles is demonstrated by evaluating triple junction geometry of polycrystalline 316L austenitic stainless steel. The validity of the measurements was established by a comparison of its result with careful manual measurement of results.

Contents

List of Tables	iii
List of Figures	iv

1 INTRODUCTION AND OBJECTIVES	1
2 MICROSTRUCTURAL CHARACTERIZATION OF POLYCRYSTALS: AN OVERVIEW	5
2.1 Characterization of Polycrystals.	5
2.1.1 Classical Stereology.	5
2.1.2 Grain boundary energy as a parameter.	11
2.1.3 Serial Sectioning and Disector Technique.	19
2.2 Digital Image Analysis.	22
2.2.1 Image Digitization.	23
2.2.2 Image Enhancement.	25
2.2.2.1 Point Operations.	25
2.2.2.2 Spatial Filtering.	27
3 MATERIALS AND MICROSCOPY	30
3.1 Material.	30
3.2 Heat Treatment.	31
3.3 Microscopy.	32
3.3.1 Specimen Preparation and Examination.	32
3.3.2 Image Analysis and Disector Thickness.	32
4 IMAGE ANALYSIS ALGORITHM AND THE DISECTOR TECHNIQUE	34
4.1 Image processing algorithm.	34
4.1.1 Binarization/ Thresholding.	34
4.1.2 Erosion filtering.	36
4.1.3 Thinning Morphological Filter.	38
4.1.4 Pruning Filter.	39
4.1.5 Bitmaps.	40
4.2 Image Analysis Algorithm: Evaluation of Plane Dihedral Angles	41
4.2.1 Triple Point Detection	41

4.2.2 Evaluation of Plane Dihedral Angles(PDA)	43
4.3 3-Dimensional Stereological Probe: The Disector.	47
4.3.1 Transformation of Plane Dihedral Angles (PDA) to True Dihedral Angles (TDA)..	47
4.3.2 Computation Methodology of True Dihedral Angle.	51
5 RESULTS AND DISCUSSIONS.	55
6 CONCLUSIONS.	67

References

Appendix

LIST OF TABLES

Table 3.1. Nominal chemical composition of 316L austenitic stainless steel	30
Table 3.2: Statistical parameters of plane dihedral angles and annealing temperatures	31
Table 5.1: Statistical parameters of plane dihedral angles	56

LIST OF FIGURES

Figure 1: An optical microstructure of a polycrystalline material. β_1 , β_2 and β_3 are the plane dihedral angles at a triple point.	2
Figure 2: Illustration of two parallel section of a polycrystal. (Murr 1975)	4
Figure 2.1: Test line thrown randomly on a polycrystalline microstructure, intersecting grain boundaries at five locations	7
Figure 2.2: Illustration of contribution of spheres to give sections of diameters d_1 to d_5	11
Figure 2.3. geometrical relationship in the intersection of a sphere by a plane with a slice h .	11
Figure 2.4a: illustration of geometry of three planes	12
Figure 2.4b: A polycrystalline structure (X) consisting of randomly oriented triple edges.	12
Figure 2c: β_1 , β_2 and β_3 are the plane dihedral angles at a triple point.	12
Figure 2.5: A random section through two grain boundary planes	17
Figure 2.6 : computer animation of part serial sections stacked together	20
Figure 2.7: Tracing of few representatives of serial sections	20
Figure 2.8: Two planes of a polycrystalline microstructure	22
Figure 2.9: A digitized image indicating array of pixels	24
Figure 2.11: Gray scale 8 image (a) of austenitic stainless steel with resulting binary image (b).	27
Figure 3.1: The disector consisting of the two parallel sections.	33

Figure 4.1: (a) A typical grayscale image of a polycrystalline microstructure; (b) Bimodal gray level histogram (c) transformed binary image (after applying thresholding algorithm)	35
Figure 4.2: illustration of erosion	36
Figure 4.3a: A typical structure element of a erosion filter.	37
Figure 4.3b: Result of applying erosion filter	37
Figure 4.5: Illustration of thinning morphological transformation.	38
Figure 4.6: Image of microstructure resulted after thinning morphological filter was applied.	38
Figure 4.7: Illustration of pruning transformation	39
Figure 4.8: Image of microstructure resulted after pruning filter was applied.	39
Figure 4.9: Bitmap of the marked rectangular region in figure 4.8.	40
Figure 4.10: A partial configurations of valid triple points.	42
Figure 4.11: Illustration of pseudo triple points	46
Figure 4.12: Tangents (thick lines) drawn to the fitted grain boundary segments	48
Figure 4.14. Microstructures of 316L austenitic stainless steel with microhardness indents as reference markers	54
Figure 5.1: Unprocessed and processed image	58
Figure 5.2: Typical example of polynomial fits	59
Figure 5.3: PDA distribution of sample	60
Figure 5.4: Illustration of standard deviation of plane dihedral angles	61
Figure 5.5: TDA distribution of sample	63
Figure 5.6: Illustration of standard deviation of true dihedral angles	64
Figure 5.7 : Illustration of standard deviation of relative grain boundary energy	65
Figure 5.8: Relative grain boundary energy distribution of sample	66

CHAPTER 1

INTRODUCTION AND OBJECTIVES

Interfaces are a fundamental constituent of the polycrystalline microstructure. The geometry of these interfaces is associated with their energy state, which is strongly linked with mechanical properties of materials (Singh 1998). In a polycrystalline space filling microstructure mating of three grains form a triple edge. The dihedral angles are the angles between any two surfaces (or interfaces) that meet at the triple edge. These angles have been termed as true dihedral angles (Murr 1975). While , in a polycrystalline microstructure observed on a two-dimensional section, the angles between grain boundary segments at a triple point are termed as plane dihedral angles. The geometry of triple junctions, described in terms of the true dihedral angles between grain boundaries, is related to the energy state of grain boundaries and consequently to the deformation behavior of polycrystals [12,14-16]. Figure 1 shows the description of triple point geometry by the angles between the grain boundary segments on a polycrystalline micrograph of a plane section.

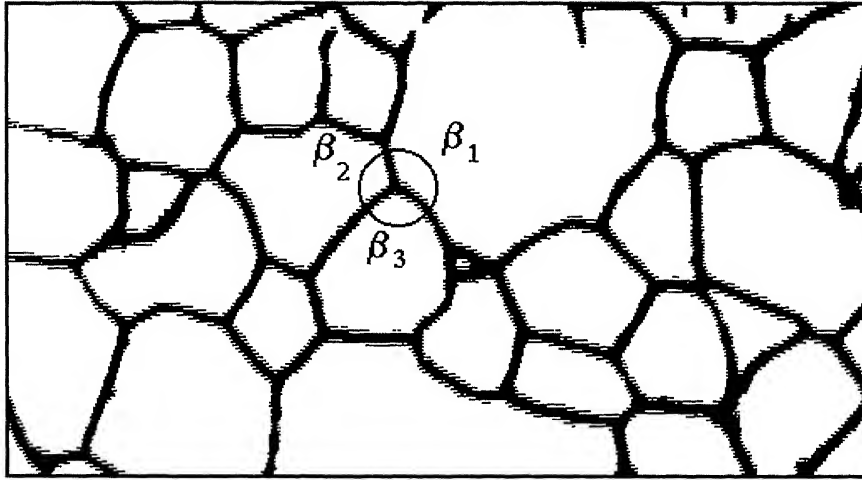


Figure 1: An optical microstructure of a polycrystalline material. β_1 , β_2 and β_3 are the plane dihedral angles at a triple point.

Even though dihedral angles measure has been considered to be an important parameter for the characterization of polycrystals, it has been used to characterize polycrystalline microstructure only to a limited extent . This is attributed to the fact that the procedures generally followed are tedious and the results may not reflect good statistical estimate of the distribution of plane dihedral angles. Therefore the main objective of this study is to develop an automated technique based on digital image processing and analysis to obtain a large and an accurate statistical estimate of the distribution of plane dihedral angles. For testing the validity of the technique it has been applied to samples of 316L austenitic stainless steel.

For investigation of triple junction geometry numerous methodologies are available among which quantitative stereology promises to be very efficient. It provides a means to describe the geometry of three-dimensional microstructure from observations made on plane two-dimensional sections. Classical stereological methods typically involve measurements made on a single plane section. The disadvantages of using a single section are well known (Gundersen 1986) and can be

completely overcome by using the technique of serial sectioning. However, the serial sectioning technique is very cumbersome and time consuming. Some of the disadvantages of both the single section and serial sectioning techniques have been overcome by the three-dimensional sampling probe termed as the “disector” (Sterio, 1984, Gundersen, 1988). The disector technique, which uses two parallel plane sections separated by known distance as shown in figure 2, has been used for an unbiased determination of numerical density, N_V (Gundersen 1986)

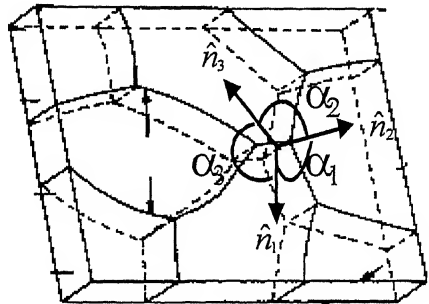


Figure 2: Illustration of two parallel section of a polycrystal and true dihedral angles.
(Murr 1975)

The other objective is to directly evaluate true dihedral angles. It is shown that a statistically significant number of measurements of true dihedral angles can be performed using automated digital image processing. Further, these measurements have been used to obtain a distribution of relative grain boundary energies.

Therefore to summarize, main objective of this study is to develop a combination of an automated technique based on digital image processing/analysis and the three-dimensional probe, disector, to obtain a large and an accurate statistical estimate of distribution of plane dihedral angles from two sections of the given sample and to transform it to true dihedral angle distributions. Further, to show that the true dihedral angles can be used for evaluating relative grain boundary energies.

CHAPTER 2

MICROSTRUCTURAL CHARACTERIZATION OF POLYCRYSTALS: AN OVERVIEW

2.1 Characterization of Polycrystals

To characterize a polycrystalline microstructure various parameters, like grain size distributions, grain volume, surface area measurements, curvatures etc, have been extensively examined by studying the two-dimensional planar sections using quantitative stereology.

2.1.1 Classical Stereology

Stereology has emerged as an effective tool to characterize a microstructure. Quantitative stereology is generalized body of methods, applicable to any material, for quantitative characterization of three-dimensional microstructures by means of two-dimensional sections taken at random from the solid material (Underwood, 1970). Much of the earlier works since has concentrated on interpretation of two-dimensional measurements obtained from a planar section and relating it to three-dimensional parameters defining the structure (Wiel, 1980).

The methods involved in stereology allow us to infer parameters of the structure from incomplete information gathered from sections of a dimension lower than that of the structure. This has two consequences; firstly certain parameters of structure (connectivity between parts) are not accessible for stereological evaluation and secondly the section being a sample of a particular type, stereological methods becomes statistical in nature and measurements obtained on these samples yield estimates, which are averages and are affected by some sampling error. This limitation requires that the unit on which the parameter is defined occur in large number within the structure. Thus it is easy to estimate the total surface, average diameter and the size distribution of a population but it is impossible to estimate the surface or the diameter of a single object by stereological methods, unless we are able to cut a large number of random sections from this one and only object. A section or probe is a statistical sample of the structure, which is independent of the geometry of the structure. The section image that we observe must be a chance intersection between structures and probe i.e. random sampling. In 1945 Saltykov showed that the boundary length of profiles on sections was proportional to the surface area of the embedded objects and this surface can be estimated from the number of intersections the boundary form with a random test line. It was also observed that the principle establishing the probabilistic relationship between the area of a surface and length of a line, expressed by the number of intersections formed when they interact in random fashion as shown in figure 2.1. It can be seen that $N_L = 5$ (number of line segments formed) and $P_L = 5$ (number of points formed by intersection of the test line and the grain boundaries), then the surface area of the grain boundary in a unit volume is given by:

$$S_V = 2P_L = 2N_L \quad (2.1)$$

This principle was also extended to estimate the length of linear features of the structures like grain boundary edges. The reference system is now a set of planes instead of test lines and the observer counts the number of times the lineal features transect the plane of defined area.

The fundamentals of stereological symbols are essentially noted as a double symbol of two capital letters. The first capital letter defines the component parameter and the second written as subscript, represents the parameter of the reference space. To illustrate the symbol; the numerical density of some particulate component (b) in the containing space (c) is written as

$$N_v = N(b)/V(c)$$

And the areal density of profiles of (b) in the profile of structure (c) is represented as

$$A_A = A(b)/A(c)$$



Figure 2.1: Test line thrown randomly on a polycrystalline microstructure, intersecting grain boundaries at five locations

DISTRIBUTIONS

Stereological principles were also developed to explicitly consider the individual object or element of structure while determining the number and size of particles. Sectional profiles are mostly smaller than the object they have been derived. Stereological methods are now available by which we can transform measurements of profile size into dimensions describing the size of the objects. These problems were first tackled by Wicksell (1925) to solve a corpuscular problem in anatomy. He derived relationship between the size distribution of spherical and ellipsoidal particles with that of their profiles on the section. Saltykov (1967) developed a method for transforming distribution of two-dimensional measurement of sections to three-dimensional distribution of spheres. The problem was complicated by the fact that spheres with different diameters $j, j+1, j+2$ etc will contribute sections of the same size 'i', so source of each section were determined. To illustrate the transformations for size distributions a polydispersed system of spheres that had been separated into five class intervals. The number of spheres per unit volume in each class interval is

$$(N_V)_1, (N_V)_2, (N_V)_3, (N_V)_4 \text{ and } (N_V)_5$$

and their corresponding radii are

$$r_1, r_2, r_3, r_4 \text{ and } r_5$$

the total number of circular sections of a given size per unit area, regardless of origin, is

$$(N_A)_1, (N_A)_2, (N_A)_3, (N_A)_4, \text{ and } (N_A)_5$$

where the section radii vary from

$$0 \text{ to } r_1, \quad r_1 \text{ to } r_2, \quad r_2 \text{ to } r_3, \quad r_3 \text{ to } r_4 \text{ and } r_4 \text{ to } r_5$$

Figure 2.2 illustrates the possibilities arising from a random plane intersecting particles of all five sizes. Spheres of the largest class interval will give sections of all radii when cut by a random plane also the observed number of sections of the smallest size per unit area, $(N_A)_1$, is sum of contributions from spheres of all sizes represented as:

$$(N_A)_1 = \sum (N_A)_{i,j} = (N_A)_{1,1} + (N_A)_{1,2} + (N_A)_{1,3} + (N_A)_{1,4} + (N_A)_{1,5} \quad (2.2)$$

where i represents constant section size and j represents the sphere diameters. The largest section on the plane of polish can unambiguously belong to the largest sphere only which is represented as:

$$\begin{aligned} (N_A)_{5,5} &= (N_A)_5 \\ (N_A)_{4,4} &= (N_A)_4 - (N_A)_{4,5} \\ (N_A)_{3,3} &= (N_A)_3 - (N_A)_{3,5} - (N_A)_{3,4} \\ (N_A)_{2,2} &= (N_A)_2 - (N_A)_{2,5} - (N_A)_{2,4} - (N_A)_{2,3} \\ (N_A)_{1,1} &= (N_A)_1 - (N_A)_{1,5} - (N_A)_{1,4} - (N_A)_{1,3} - (N_A)_{1,2} \end{aligned} \quad (2.3)$$

$(N_A)_5$, is measured from the test plane , $(N_A)_{5,5}$ is known

Further in order to deduce how many sections of each size come from spheres in each class interval, probabilities of random planes intersecting a sphere to give sections of each size. If P_{ij} is the probability of a test plane intersecting a sphere within a slice of height h, (as shown in figure 2.3) is given by:

$$P_{ij} = h / r_{\max} = h_{i-1} - h_i / r_{\max} \quad (2.4)$$

$$P_{ij} = \frac{1}{r_{\max}} [\sqrt{(r_{\max})^2 - (r_{i-1})^2} - \sqrt{(r_{\max})^2 - (r_i)^2}]. \quad (2.5)$$

if the polydispersed system of spheres is divided into class intervals then,

$$(N_V)_j = \sum_i (N_A)_{i,j} / D_j \quad mm^{-3}$$

$$\sum_j (N_A)_{i,j} = P_{ij} D_j (N_V)_j \quad (2.6)$$

$(N_A)_5$, is measured from the test plane, therefore $(N_A)_{5,5}$ is known from equations 2.3. P_{ij} , is calculated from equation 2.5; D_5 is measured from the plane of polish. $(N_V)_5$ is thus evaluated from equation 2.6. Knowing $(N_V)_5$ we proceed to calculate $(N_V)_4$, from equation 2.5 probability of hitting size-5 sphere to get a size-4 section. This value of $P_{4,5}$ in equation 2.6 gives $(N_A)_{4,5}$, which when subtracted from the measured value $(N_A)_4$, gives $(N_A)_{4,4}$ from equation 2.3. Using the appropriate probability for hitting a size-4 sphere to get a size-4 section and knowing $(N_A)_{4,4}$, $(N_V)_4$ is calculated using equation 2.6. These sequences of calculations are continued down to the smallest particle size. It is seen that each subtracted term depends upon the previously calculated values of number density, and that the number of the smallest particle maybe inaccurate because of accumulated errors.

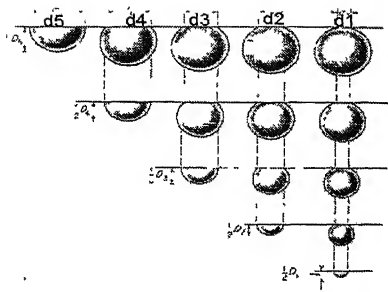


Figure 2.2: Illustration of contribution of spheres to give sections of diameters d_1 to d_5

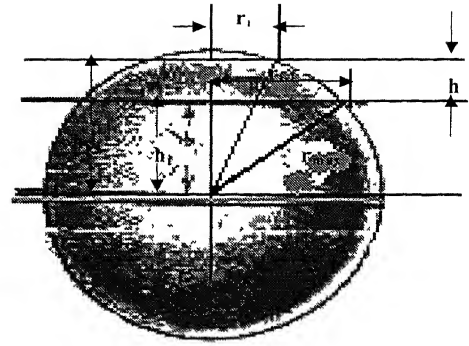


Figure 2.3: geometrical relationship in the intersection of a sphere by a plane with a slice h .

2.1.2 Grain boundary energy as a parameter

Another parameter to characterize polycrystals is grain boundary energy associated with the geometry of the junctions where three grain boundary planes meet. This parameter can be used to determine properties like plastic deformation of any polycrystalline material (Singh and Sangal, 1998). The grain boundary energy distribution has been investigated by Phumphery et al. (1974) and Varin et al. (1980) by studying the spreading and disappearance kinetics of grain boundary locations in TEM samples. However the technique is cumbersome and prone to error in the estimation temperature and time of disappearance of EGDB. Murr et al. has evaluated grain boundary energy ratio by

directly measuring true dihedral angles using TEM images in which the three dimensional structures of a grain boundary planes meeting at triple edges can be observed as shown in figure 2.4a. In a polycrystalline microstructure, the plane dihedral angles are the angles at the triplepoints as indicated in figure 2.4c. Even though dihedral angles have been considered to be important parameters for the characterization of polycrystals, they have been used to characterize polycrystalline microstructure only to a limited extent. This is attributed to the fact that the procedures generally followed are tedious and the results may not reflect good statistical estimate of distribution of plane dihedral angles. Sangal (1998) has evolved alternate approaches to estimate distribution of true dihedral angles more efficiently from plane dihedral angles using rigorous stereological analysis.

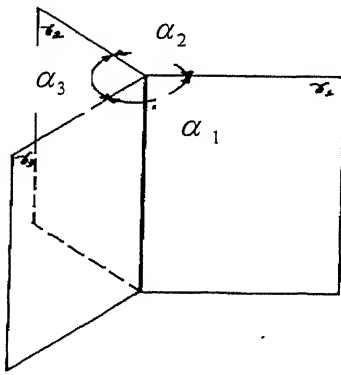


Figure 2.4a: illustration of geometry of three planes of the grains meeting at common junction.

α_1, α_2 and α_3 are the set of dihedral angles.

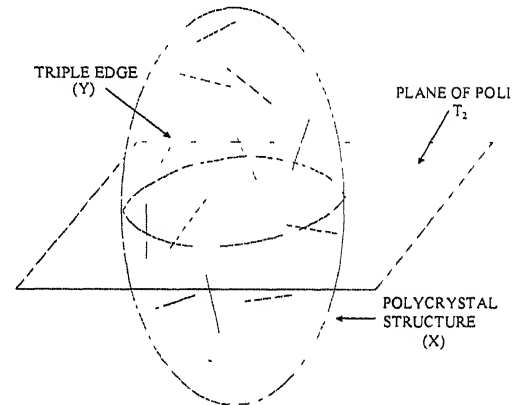


Figure 2.4b: A polycrystalline structure (X) consisting of randomly oriented triple edges being sectioned by the plane of polish (T_2).

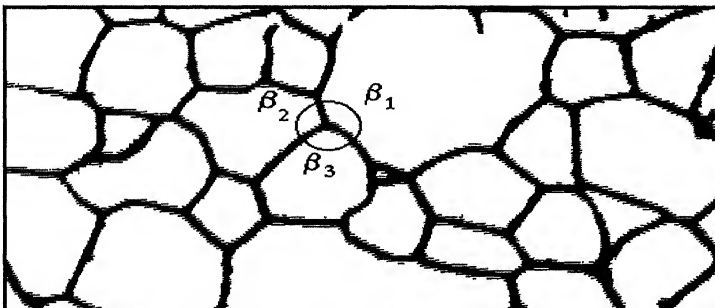


Figure 2c: β_1, β_2 and β_3 are the plane dihedral angles at a triple point.

Estimation of true dihedral angle (TDA) distribution

In a polycrystals the grain boundaries may have different values of energy, which may lead to difference in distribution of TDA (angle between the grain boundary planes meeting at a common edge, known as triple edge). TDA can be measured as mentioned above by TEM (transmission electron microscope) but were found to be very tedious and it is not possible to estimate statistical distribution of TDA. In practice the angles are measured as plane dihedral angles (PDA) between grain boundary segments meeting at a triple point. In a polycrystal, the triple edges are randomly oriented with respect to the plane of polish as shown in figure 2.4b. Therefore, the distribution of PDA has two components (1) variation in the TDA distribution and (2) statistical variation due to random orientation of the sectioning plane. The statistical variation tends to overshadow the actual variation in the TDA distribution. Hence such a distribution of PDA cannot be directly used for the calculation of relative energy of the grain boundaries. On the other hand, these surface measurements are relatively easy and it is possible to make a large number of measurements in a short time. Therefore, in the following sections, a transformation procedure is developed to unfold the distribution of true dihedral angles from the measured distribution of plane dihedral angles.

Relation between TDA and PDA:

Figure 2.5 shows a randomly oriented triple edge sectioned by the plane of polish, P. the true dihedral angle between the grain boundary planes A and B is α and the corresponding grain boundary segments a and b make a plane dihedral angle of β on the plane of polish as shown in figure 2.5. The triple edge is parallel to the z-axis and the grain boundary plane A is in the XZ plane. The orientation of the plane of polish is described by the orientation of the plane normal (denoted by \hat{n}) in terms of angle ϕ and θ in spherical coordinates. The unit vectors \hat{r}_1 and \hat{r}_2 are normal to the grain boundary planes A and B respectively. Let \vec{a} and \vec{b} be the vectors parallel to the grain boundary segments 'a' and 'b' respectively.

From the figure 2.5, the vectors \hat{r}_1, \hat{r}_2 and \hat{n} can be expressed as follows:

$$\hat{r}_1 = \hat{j} \quad (2.7)$$

$$\hat{r}_2 = -\sin \alpha \hat{i} + \cos \alpha \hat{j} \quad (2.8)$$

$$\hat{n} = \sin \theta \cos \phi \hat{i} + \sin \theta \sin \phi \hat{j} + \cos \theta \hat{k} \quad (2.9)$$

since \vec{a} is the line of intersection of planes A and P, while \vec{b} is the line of intersection of planes B and P, \vec{a} and \vec{b} can be represented by the following cross-products:

$$\vec{a} = \hat{r}_1 \times \hat{n} \quad (2.10)$$

$$\vec{b} = \hat{r}_2 \times \hat{n} \quad (2.11)$$

from the equations 2.7 to 2.11, \vec{a} and \vec{b} can be expressed in the following form,

$$\vec{a} = \cos \theta \hat{i} - \sin \theta \cos \phi \hat{k} \quad (2.12)$$

$$\vec{b} = \cos \alpha \cos \theta \hat{i} + \sin \alpha \cos \theta \hat{j} - \sin \theta \cos(\phi - \alpha) \hat{k} \quad (2.13)$$

β , the plane dihedral angle, can be expressed as:

$$\cos \beta = \frac{\vec{a} \cdot \vec{b}}{|\vec{a}| \cdot |\vec{b}|} \quad (2.14)$$

The terms in above equation can be obtained from equations 2.12, 2.13 and 2.14:

$$\vec{a} \cdot \vec{b} = (\cos \theta \hat{i} - \sin \theta \cos \phi \hat{k})(\cos \alpha \cos \theta \hat{i} + \sin \alpha \cos \theta \vec{j} - \sin \theta \cos(\phi - \alpha) \hat{k})$$

$$\vec{a} \cdot \vec{b} = \cos^2 \theta \cos \alpha + \sin^2 \theta \cos \phi \cos(\phi - \alpha) \quad (2.15)$$

$$|\vec{a}| = \sqrt{\cos^2 \theta + \sin^2 \theta \cos^2 \phi} = \sqrt{1 - \sin^2 \theta \sin^2 \phi} \quad (2.16)$$

$$|\vec{b}| = \sqrt{\cos^2 \alpha \cos^2 \theta + \sin^2 \alpha \cos^2 \theta + \sin^2 \theta \cos^2(\phi - \alpha)} = \sqrt{1 - \sin^2 \theta \sin^2(\phi - \alpha)} \quad (2.17)$$

Thus from equations 2.8 to 2.11, the following expression can be obtained

$$\cos \beta = \frac{\cos^2 \theta \cos \alpha + \sin^2 \theta \cos \phi \cos(\phi - \alpha)}{\sqrt{(1 - \sin^2 \theta \sin^2 \phi)(1 - \sin^2 \theta \sin^2(\phi - \alpha))}} \quad (2.18)$$

It is seen from equation 2.12 that if $\theta = 0$ then $\beta = \alpha$, as would be expected. However, for any given value of α , β can vary from 0° to 180° depending upon the orientation of the plane of polish.

On sectioning the polycrystalline structure (set X), the probability of an intersection of a triple edge (set Y) with the sectioning plane is given by (Sangal, 1998):

$$P_1(T_2 \cap Y | T_2 \uparrow X) = \frac{2l}{\pi} \quad (2.19)$$

where l is the mean length of a triple edge. Consider a set Y_α consisting of pairs of intersecting planes making equal angles α distributed in the polycrystalline space. The intersection of Y with the sectioning plane, T_2 will give rise to a probability distribution of plane dihedral angles given by $p(\beta/\alpha)d\beta$. Let the numerical density of the set Y_α be $N_V(\alpha)d\alpha$, i.e. number of plane pairs making an angle of α per unit volume. The probability density, $g(\beta)$ of plane dihedral angles is given by

$$g(\beta) = \frac{\int P(T_2 \cap Y | T_2 \uparrow X) p(\beta/\alpha) N_V(\alpha) d\alpha}{\int_\alpha P(T_2 \cap Y | T_2 \uparrow X) N_V(\alpha) d\alpha} \quad (2.20)$$

Substituting $P_1(T_2 \cap Y | T_2 \uparrow X)$ from equation 2.19, the probability density $g(\beta)$ reduces to :

$$g(\beta) = \int_\alpha p(\beta\alpha) f(\alpha) d\alpha \quad (2.21)$$

Where $f(\alpha) = N_V(\alpha) / \int_\alpha N_V(\alpha) d\alpha$ is the probability density function of TDA. Equation 2.21, which is an integral equation, relates the PDA distribution to the TDA distribution. In order to solve equation 2.21, it is discretized by considering discrete distributions of plane as well as dihedral angles. Both the distributions are divided into several classes. The integral equation 2.21 is converted into set of simultaneous equations:

$$\begin{vmatrix} g_1 \\ \vdots \\ g_j \\ \vdots \end{vmatrix} = \begin{vmatrix} p_{11} & \cdots & p_{i1} & \cdots \\ p_{12} & \cdots & p_{i2} & \cdots \\ \vdots & & \vdots & \\ p_{1j} & \vdots & \vdots & \vdots \\ \vdots & & \vdots & \end{vmatrix} \begin{vmatrix} f_1 \\ f_2 \\ \vdots \\ f_i \\ \vdots \end{vmatrix} \quad (2.22)$$

Where, g_j is the frequency of the j^{th} class of plane dihedral angles; f is the frequency of the i^{th} class of true dihedral angles; p is the probability of generating plane dihedral angle in j^{th} class from a true dihedral angle in the i^{th} class. A Monte Carlo procedure has been developed for the estimation of coefficient p_{ij} . For different class widths, a new set of values for these coefficients has to be calculated. The frequencies g_j can be determined by the measurement of PDA on the polycrystalline microstructures. Now the frequencies of TDA can be calculated by solving the simultaneous equations given in relation 2.22.

2.1.3 Serial Sectioning and Disector Technique

Test probes of fewer than three dimensions sample a particle in a microstructure with a chance proportional to their heights for a two dimensional probe, test lines intersect particles depending on their surfaces and test points sample particles proportional to particle volume. This method introduces a bias in the results and becomes an indirect way to count number of particles. Thus three-dimensional test probes were introduced to overcome the statistical errors caused due to the bias mentioned above. Measurement of average grain volume and certain topological parameters have been attempted by using three-dimensional serial section analysis by Rhines et al. (1976).

Serial sections consist of consecutive slices cut through a piece of material at intervals that are so narrow such that every significant detail of internal structure are exposed as shown in figure 2.6. For topological measurements it is not necessary that the slices are parallel to each other, but for determining the total volume of material that has been sectioned it is essential that the sections are flat and parallel to each other. The total volume included in the serial sections, thus produced is calculated by making micrometer

measurements of the total thickness of the sample at the first and the last section. The number of features are counted in the sectioned material is then divided by the measured volume to obtain N_v (number of features per unit volume). Not only precision is to be maintained while making successive sections but also care is to be exercised while doing etching, repeated etching on a section can systematically alter the surface contour and may alter the quantitative observations. The counting of features on section is done by numbering each grain in polycrystals as shown in figure 2.7. Additionally it is recommended to do systematic bookkeeping to record the data in usable form. The amount of effort required for the entire procedure to be successful is enormous and fairly time consuming.

By serial section method entire grain in a polycrystal can be reconstructed and true dihedral angles can also be evaluated. However this procedure was prone to many errors while tagging the series of sections accurately in an opaque sample. This procedure turned out to be extremely time consuming.

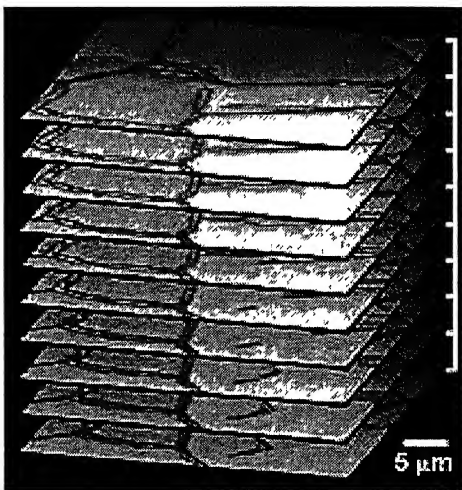


Figure 2.6 : computer animation of part serial sections stacked together.

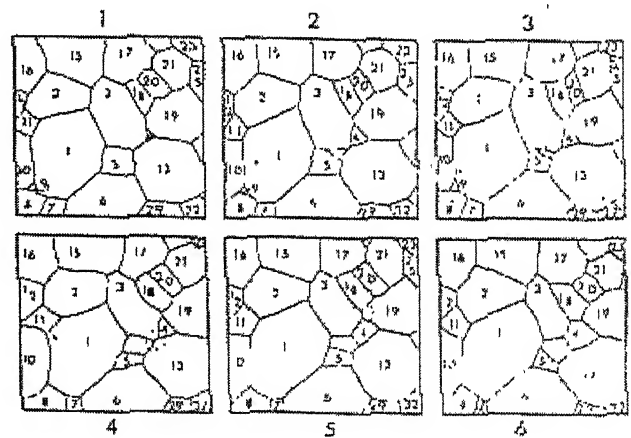


Figure 2.7: Tracing of few representatives of serial sections.

Disector

Sterio (1984) introduced the disector technique to overcome above limitations. In this technique two parallel 2-dimensional sections separated by a known thickness, h , are made on the sample and the counting is done as per requirement on these two sections. The thickness is kept smaller than the smallest particle size in the microstructure. The main advantage of disector is that nothing is assumed about the particles shape size or volume and actual counts are performed in the sample. Although the information obtained by this method is lower than serial sectioning but the overall efficiency increases multifold, Gundersen (1986). The number of features appearing on one section and not on the other is counted which is divided by volume of the disector to give N_v (volume will be disector depth multiplied by the area of the section). This technique however demands accurate calculations of the disector depth, which is discussed in the following chapters.

Dihedral angles

Measurement of the geometry of triple junctions between grain boundaries in polycrystalline materials generates large sets of dihedral angles from which maps of the grain boundary energy may be extracted as shown in the figure 2.8, Adams et al. (1999). An extended form of orientation imaging microscopy (OIM) was used to measure both triple junction geometry via image analysis in the SEM and local grain orientation via electron back scatter diffraction. Serial sectioning with registry of both in-plane images and successive sections characterizes triple junction tangents from which true dihedral angles are calculated. The procedure gave an estimate of grain boundary energy but it turned out to be extremely tedious and inefficient due to serial sectioning but image processing introduced some ease in the evaluation.

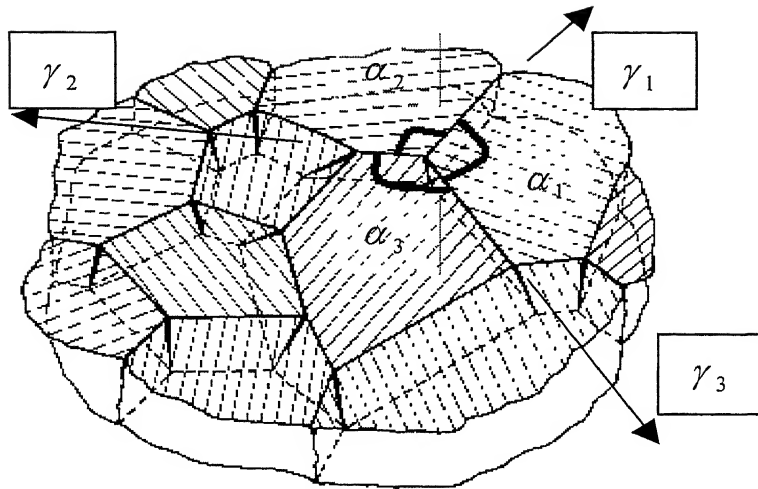


Figure 2.8: Two planes of a polycrystalline microstructure indicating dihedral angles and grain boundary energy. (Murr, L.E).

2.2 Digital Image Analysis

The practical application of stereological methods depends upon the development of adequate technology. In 1916 Shand developed the first scanning stage for a light microscope in which the specimen was driven across the microscope field containing a hair line cross of two co-linear micrometers to measure lineal distances to measure volume fraction of a particular phase. Automatic methods of image analysis were introduced in quantitative analysis by recording a field under microscope with sensitive photo detectors. These detectors record image on the basis of contrast differences. A more powerful variant called the flying spot microscope was developed by Roberts in 1951. Here a fine spot of light produced by a cathode ray oscilloscope was focused on the specimen and the image was rapidly scanned. A photomultiplier picked up the signal

emitted from the specimen and transferred to second cathode ray tube which was synchronized with the first tube.

In 1967, TV cameras were introduced to cut up the microscope image into a dense lattice of fine lines or points. Computers and image processing algorithms hence provided powerful tools to examine microstructures. It primarily acts as tool to automate the procedure of determining the evaluation of tedious stereological parameters like dihedral angles. Digital image processing have been used by many investigators in the field of quantitative characterization of multi phase materials like Silva et al. (1998) and Latala et al. (2001). They have concluded that the results obtained by computer-aided analysis provide highly consistent results. Though computer aided processing takes somewhat longer time to set up various parameters but then due to fully automated further analysis the total analysis period was about 100 times shorter than manual computations. Gokhale et al (1999), developed computer based techniques to determine grain size distribution as well as grain volume determination using serial sectioning and large area disector probes.

2.2.1 Image Digitization

An image is visual representation of an object, and image processing manipulates information within an image to make it more useful. In order to process an image with a computer the image must be converted into numeric form, this process is termed as image digitization. The digitization process divides an image into horizontal grid of very small regions called picture elements or pixels. In computer thus the whole image is represented by this digital grid. This grid is more generally called as 'bitmap'. Each pixel in the bitmap is identified by its position in the grid by it's row (x) and column number (y). The convention used (in this study) is that the (0,0) pixel is located at the upper left position of the bitmap as shown in the figure 2.9. when the source sample is scanned under a CCD camera it is examined in a grid fashion. That is each pixel in the image is individually sampled and the brightness is measured and quatified. This measurement results in a value for a pixel, for a gray scale 8 image, it will range from 0 – 255. Gray

scale-8 denotes that 8 bits are used to store each pixel value, which is also termed as the pixel depth. Depending upon the number of bits used for storing pixel value, the image can be classified as Gray Scale 8, Gray Scale 12 or Gray Scale 16. a totally black and white image is represented by 1 bit only, either the pixel value is '0' completely black or the pixel value is '1' completely white. In an 8 bit Gray scale image, a pixel with value of zero is completely black and a pixel with a value of 255 is completely white. A value of 127 represents a gray color halfway between black and white (medium-gray), and a pixel value of 64 has a color halfway between medium gray and black.

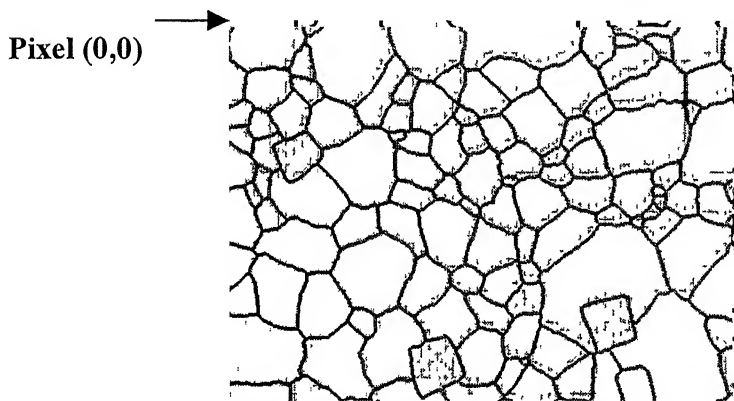


Figure 2.9: A digitized image indicating array of pixels, the pixels in the above microstructure are shown much larger than the actual size.

2.2.2 Image Enhancement

The acquired image requires some form of enhancement to improve its appearance and to extract useful data from it. Enhancement techniques range from simple operations such as brightness and contrast adjustment to more complicated such as spatial and morphological filtering operations. These operations are able to refine the visual information as per the need. The image enhancement techniques are broadly classified under following categories:

- a) Modify the intensity values of the pixels, or Point Operations
- b) Application of spatial filters
- c) Manipulate the image frequencies

Above enhancement process can be applied individually or in combo to achieve the desired results.

2.2.2.1 Point Operations

The image can be enhanced to a limited extent by modifying the intensity values of the pixels. If the image is very dark overall, all values of the pixels can be increased by stipulated amount to make it better. Brightness, contrast and thresholding are primary operations, which work by altering the intensity values of the pixels. These transformations are termed as point operations as the result of any transformation of any pixel depends only on the initial gray value of this pixel and is independent of its neighbors. When brightness is increased the value of each pixel of the image is increased closer to value '255' in a Gray Scale 8 image. When brightness is decreased the pixel intensities are reduced, moving them closer to 0, or black.

Contrast denotes the degree of difference between the brightest and darkest components of an image. An image has poor contrast if it contains only harsh black and white transitions or contains pixel values within a narrow range. An image has good contrast if it is composed of a wide range of brightness values from black to white. Increasing the contrast spreads the pixel values across a wider range, while decreasing contrast compresses the values into a narrower range.

Thresholding allows an image to be converted into just two colors: black and white. This is done by specifying a range of intensities to be emphasized, they are set to white and converting all others to black thresholding is used for segmenting an image to extract important features for further measurements or to reduce an image to two intensity levels in preparation for filtering operations.

Binarization

Currently gray images are most frequently used aid for recording image data and processing it. Binarization is an important step in image analysis (Wojnar, 1998) as the binary images occupies approximately 8 times smaller memory than gray images and almost 24 times shorter than a colored image also only in binary images can one detect separate features such as grains (every connected set of pixels is recognized as a single particle). These separated objects can now be subjected to measurements such as area, perimeter, diameter etc. another advantage of binary image is that some transformations mainly morphological operations can be done on such images only. There are procedures to separate connected particles, to reduce grain boundaries to one pixel thin lines etc falling under this category. The process of transformation of gray scale images to binary images is called binarization or thresholding as shown in figure 2.10.



Figure 2.10: Gray scale 8 image (a) of austenitic stainless steel with resulting binary image (b).

2.2.2.2 Spatial Filtering

Filtering operations reduce or increase the rate of change that occurs in the intensity transitions within an image. Areas, in which there are sudden changes in intensity, appear as distinct edges in the image. Area where there are gradual changes shows as soft edges. Filtering acts to detect and modify the rate of change at these edges. It can increase the intensity differences in a soft image to make it sharper or reduce the intensity differences in a hard edge to smooth and soften it.

Filtering operations produce their effect by modifying a pixel's value based upon the values of pixel's surrounding it. For example, blurring is accomplished by averaging all of pixel values and replacing the center pixel with the averaged value. This reduces the variations among neighboring pixels. Filtering techniques are divided into two categories: convolution and nonconvolution filters. Both techniques accomplish their results by examining and processing the image in small regions called pixel neighborhoods. A neighborhood is a square region of image pixels, typically 3X3, 5X5 or 7X7 in size.

Convolution filters

Enhancement or convolution filters process image neighborhoods by multiplying the values within a neighborhood by a matrix of filtering coefficients. This matrix is called a kernel. It is the same size as the neighborhood that it is being applied to. The results of this multiplication are summed and divided by the sum of the filter kernel. The result replaces the center pixel (sometimes also called origin), in the image neighborhood. The convolution process always uses a neighborhoods original pixel values as input. For example the kernel focus is moved to say, pixel 6 (on the image), the filtering process will use pixel 5's original value, not the one it was just assigned by convolution. The various filters under this category are low-pass, hi-pass, sharpen median etc.

Non-Convolution/ Morphological filters

Morphological filters are highly abstract theory of image transformations, possessing it's own rules and notations (Wojnar, 1998). Morphological processing refers to certain operations where an object is hit with a structuring element (kernel) and thereby reduced to a more revealing shape. The structuring element are defined using following notation:

- 1- represents pixels belonging to the set of points under analysis for example black points in figure 2.11(b).
- 0- represents pixels belonging to the matrix for example white points in figure 2.11(b)
- X- represents pixels not taken into account, that is, this point can have any value and has no effect in the transformation.

Non-convolution filters also work with pixel neighborhoods but do not multiply the neighborhood values by a kernel like in convolution filter. This filter works with the data

in the neighborhood itself and uses either a statistical method or a mathematic formula to modify the pixel upon which it is focused. Erosion , dilation, open and close are few examples of morphological filters, which is described in detail in section (experimental procedure).

CHAPTER 3

MATERIALS AND MICROSCOPY

3.1 MATERIAL

The experiments were performed on commercial polycrystalline 316L austenitic stainless steel. The chemical composition of the steel (determined using JEOL electron probe micro analyzer) is given in Table 1. The material, in the form of 12.5 mm thick plates, was cold rolled to 92% reduction.

Table 3.1: Nominal chemical composition of 316L austenitic stainless steel.

Elements	Si	Ni	Ti	W	C	Cr	Mo	Cu	Al	Fe
Composition %	0.528	13.73	0.029	0.076	0.003	18.05	2.025	0.019	0.036	balance

3.2 HEAT TREATMENT

The cold rolled material was given five different annealing treatments as given in the Table 2. The heat treatment was carried out in a Kanthol furnace. Small pieces of rolled samples were annealed at five different temperatures. The Table also gives the variation in grain size (measured as mean intercept length). The method of determining grain size is discussed in section 2.1.1.

Table 3 2: Statistical parameters of plane dihedral angles and annealing temperatures

ANNEALING TEMP ($^{\circ}C$)	ANNEALING TIME (Minutes)	GRAIN SIZE* (μm)	DISECTOR THICKNESS $h(\mu m)$
900	60	24.916	3.80
900	180	26.69	4.50
950	60	35.93	3.18
1000	60	53.39	6.16
1050	60	62.29	6.25

* mean intercept length has been taken
as a measure of grain size

3.3 Microscopy

3.3.1 Specimen Preparation and Examination

For accurate automated measurements of microstructural parameters sample preparation becomes critical. Mechanically polished samples were electrochemically etched: 60% HNO₃ was used as electrolyte; electrode used was 316L stainless steel; current density was kept at 0.75 mA/mm². This etching procedure ensures that the twin boundaries are not revealed. Another advantage is that it preferentially etches the grain boundaries resulting in a sharp black and white contrast. Such a contrast is ideal for automatic image analysis.

3.3.2 Image Analysis and Disector Thickness

24 bit RGB digital images were acquired through a CCD camera mounted on Lieca Optical Microscope. These images were subsequently converted to 8-bit grayscale images. The size of the acquired image was 640X480 picture elements. A micro-scale was used to perform calibration of the image. At a magnification of 100X, dimension of one pixel was calculated to be 0.6041 μm . The software used was Image-Pro Plus ver 4.1 for capturing as well as processing the images.

After polishing and etching the first section of the disector, several micro hardness indents were put as reference markers using a load of 160 gms by a Leica micro hardness indenter. The indenter used was a standard diamond pyramid having an angle of 136° between its two opposite faces. Digital image processing was employed in order to accurately measure microstructural parameters and the positions and sizes of the hardness indents. This was followed by mechanical re-polishing and re-etching to produce the second section (parallel to the first

section) of the disector. The hardness indentations were used to align the images of the two sections as shown in figure 3.1. The decrease in the size of the indents was used to establish the separation distance, h , of the two sections S_1 and S_2 as shown in figure 1. The distance of separation, h , which is termed as the disector thickness, is given by :

$$h = (D_1 - D_2) / 7.0006 \quad (3.3.2.1)$$

where, D_1 and D_2 are the average diagonal lengths of indents in sections S_1 and S_2 respectively

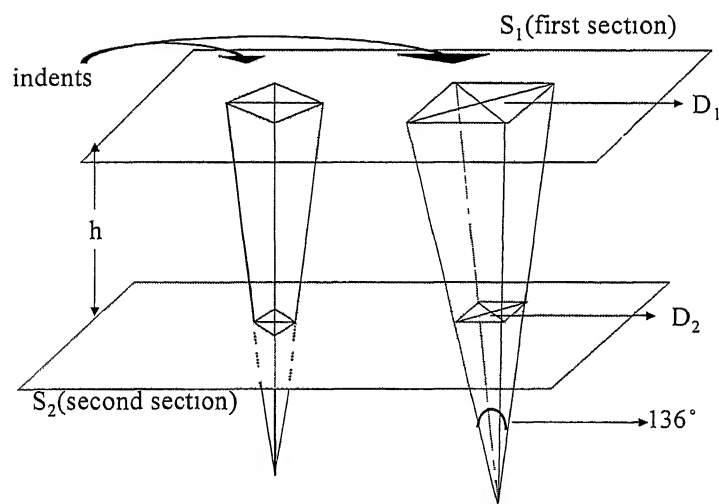


Figure 3.1: The disector consisting of the two parallel sections S_1 and S_2 separated by a distance h . Indents are produced by a standard diamond pyramid indenter of a microhardness tester. D_1 and D_2 are the diagonal lengths of the indents.

CHAPTER 4

IMAGE ANALYSIS ALGORITHM AND THE DISECTOR TECHNIQUE

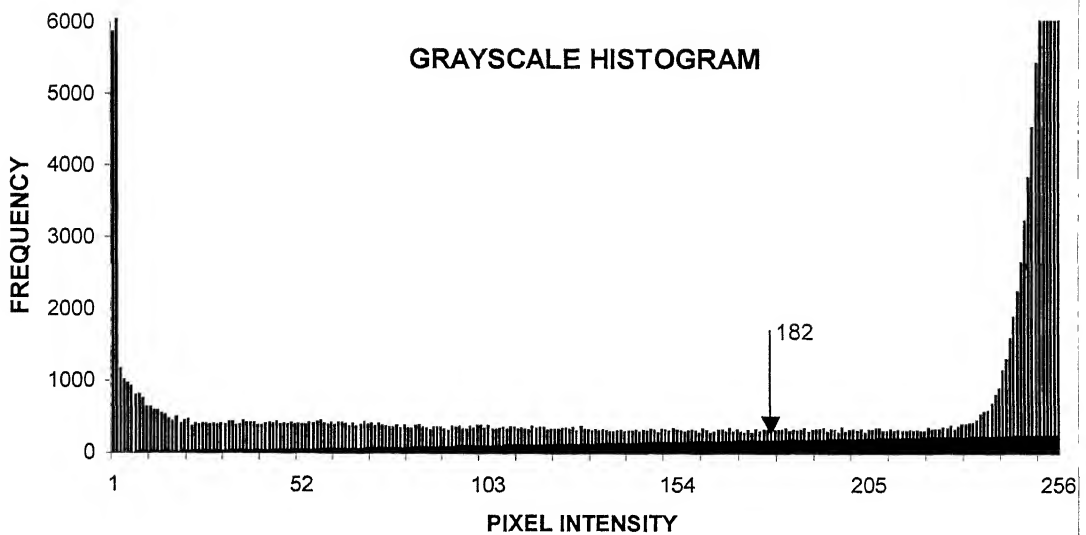
4.1 Image processing algorithm

4.1.1 Binarization/ Thresholding

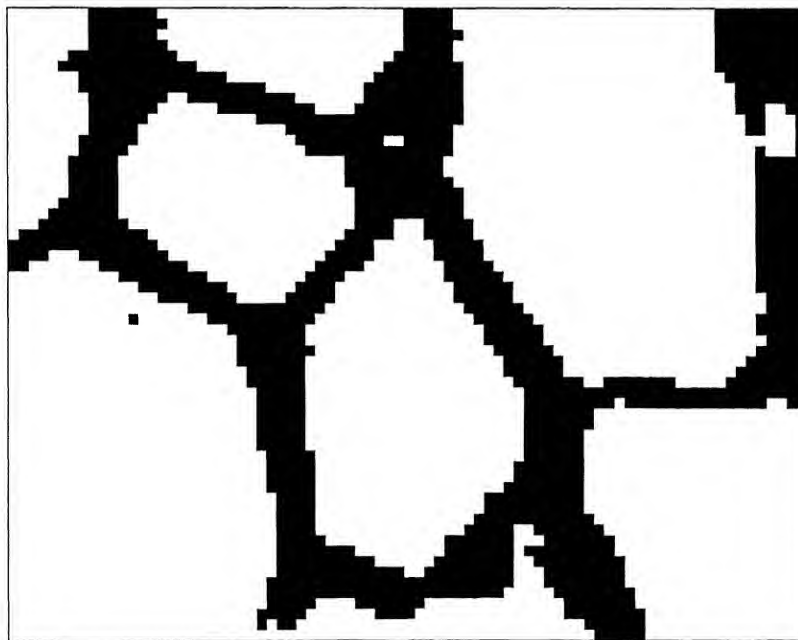
A typical grayscale image of the microstructure of 316L austenitic stainless steel and the corresponding bimodal gray level distribution are shown in figures 4.1(a) and 4.2(b) respectively. For the binarization of such images as mentioned in section 2.2, automatic thresholding was applied, which involved fixing the threshold limit at a local minimum between two local maxima of gray level distribution. All pixel values below the threshold limit were set to '0' value and those pixels whose value was above this limit were fixed to '255' value. This type of binarisation gave consistent results independent of image contrast and brightness. Figure 4.1(c) shows the resulting binary image in which all the sensor noise and background illumination variation have been abstracted.



(a)



(b)



(c)

Figure 4.1: (a) A typical grayscale image of a polycrystalline microstructure; (b) Bimodal gray level histogram of the image in figure (a), arrow mark indicates thresholding limit; (c) transformed binary image (after applying thresholding algorithm) showing polishing/etching artifacts

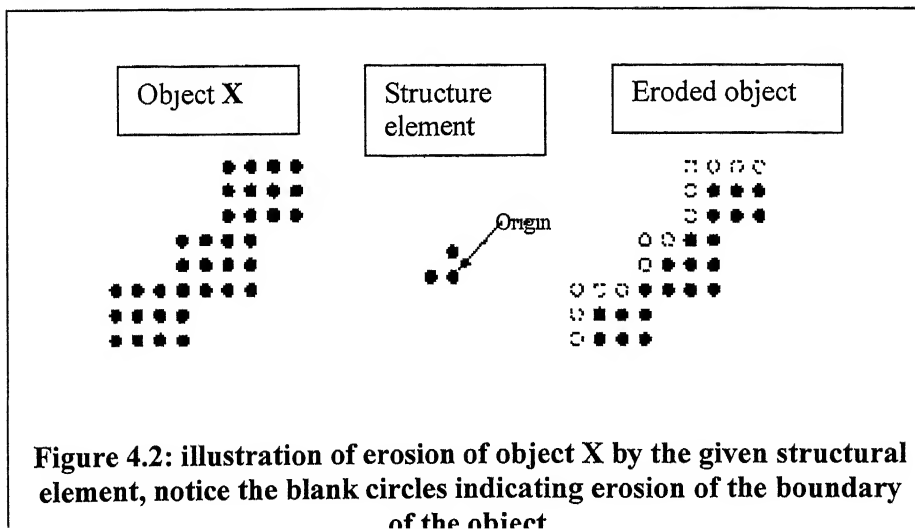
4.1.2 Erosion filtering

The erosion filter is a morphological filter that changes the shape of an object in an image by eroding or reducing the boundaries of the specified objects. Suppose the object X and the structuring element B are represented as sets in Euclidean space. Let B_x represent translation of B so that its origin is located at x . then the erosion of X by B is defined as the set of all points x such that B_x is included in X :

$$X(-)B \triangleq \{x : B_x \subset X\} \quad (3.1)$$

Erosion is a shrinking filter therefore eroded objects are smaller than the initial one. Some narrow peninsulas or small particles will disappear after erosion. Figure 4.2 shows example of erosion filter.

This filter removes noise/artifacts created by etch pits, inhomogeneous etching/polishing etc, in the image, erosion filter was applied which removed all the pixels touching the matrix. This operation resulted in the removal of some narrow peninsulas and small artifacts as shown in figure 4.3. A typical erosion kernel is shown in figure 4.3(a), this kernel or the structuring element fits with the configurations on the image pixels and does transformation only to the pixels which matches exactly with this structuring element. It is often used to reduce , or eliminate small bright objects as this filters erodes the boundaries of bright objects and enlarges the boundary of the dark ones. The image shown in figure 4.3b has been inverted to make convenient comparison with the original image as shown in figure 4.1 (a).



X	1
1	1 origin

Figure 4.3a: A typical structure element of a erosion filter.

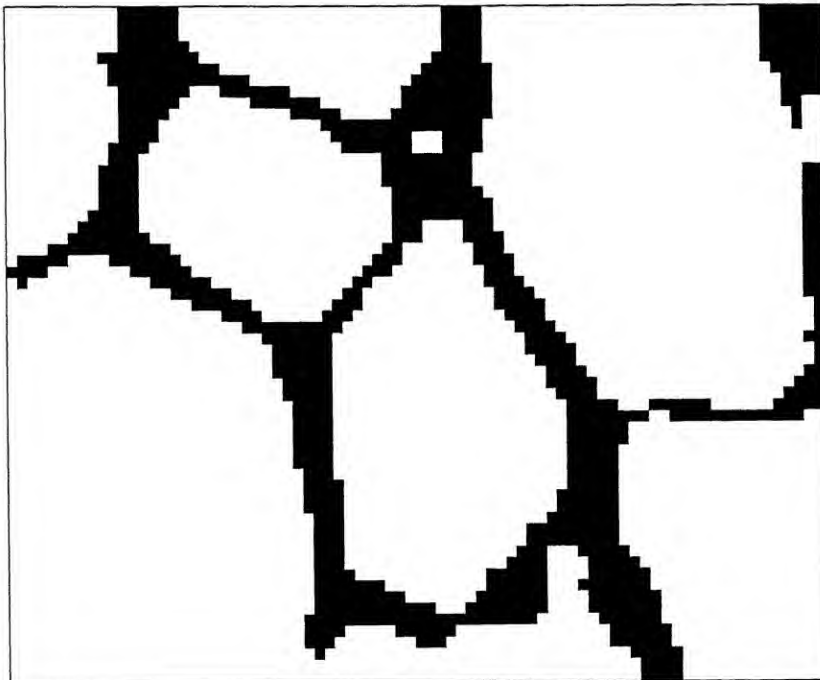


Figure 4.3b: Result of applying erosion filter on figure 4.1(c). notice the disappearance of peninsulas and reduction in thickness of grain boundaries.

4.1.3 Thinning Morphological Filter

The eroded image is transformed by a thinning morphological operation to reduce the grain boundaries to only one pixel thick line. The thinning filter was applied iteratively till further operations of the filter had no effect on the image. The filter's basic operation is as shown in figure 4.5. The resulting thinned image is shown in figure 4.6. However, this image had numerous open-ended branches, which appear because of non-homogeneous etching. Such branches can lead to an erroneous detection of triple points.

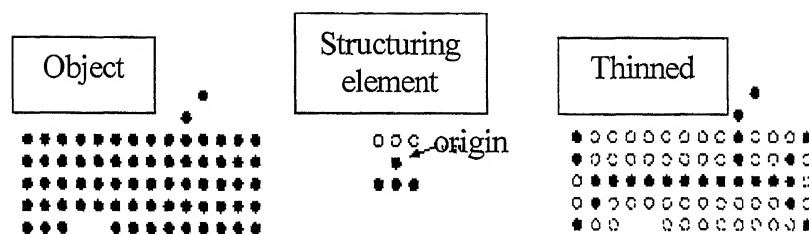


Figure 4.5: Illustration of thinning morphological transformation of the object, notice the reduction of the object to one unit thick line.

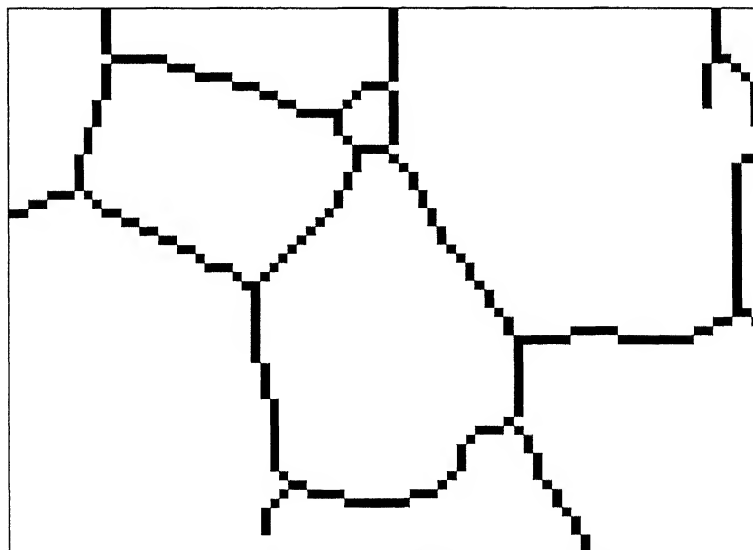


Figure 4.6: Image of microstructure resulted after thinning morphological filter was applied on image shown in figure 3.3, notice that the grain boundaries now are only one pixel thick.

4.1.4 Pruning Filter

Therefore to smoothen the thinned image and to remove all parasitic branches a pruning filter was applied. Figures 4.7 and 4.8 show the basic pruning operation on the object and the final pruned image respectively.

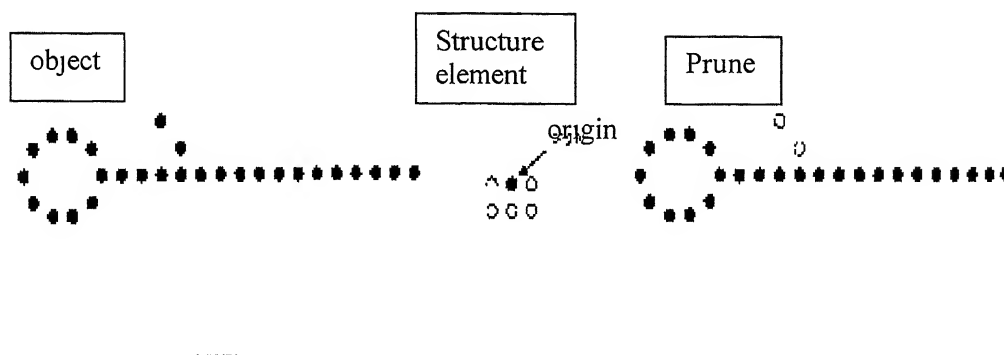


Figure 4.7: Illustration of pruning transformation of the object, notice that the parasitic branch has disappeared.

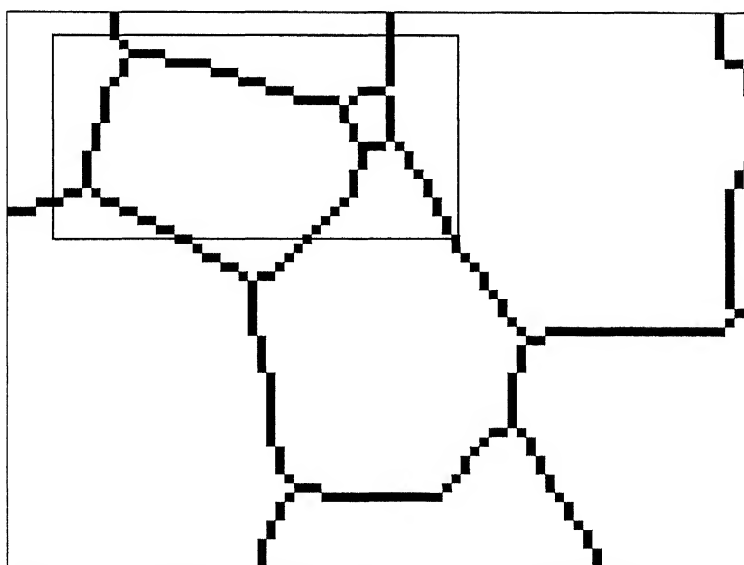


Figure 4.8: Image of microstructure resulted after pruning filter was applied on image shown in figure 4.6, notice that the grain boundaries now are only one pixel thick.

4.1.5 Bitmaps

The final image was converted into a bitmap. Figure 4.9 shows a bitmap of the rectangular region marked in figure 4.8. As can be seen from this figure, the grain boundary pixels are represented by 0's and the matrix pixels are represented by 1's (these pixels uniquely had a value of 255 in the binary image). The bitmap was used as input to the algorithm developed to detect triple junctions and determine their geometry, as discussed below.

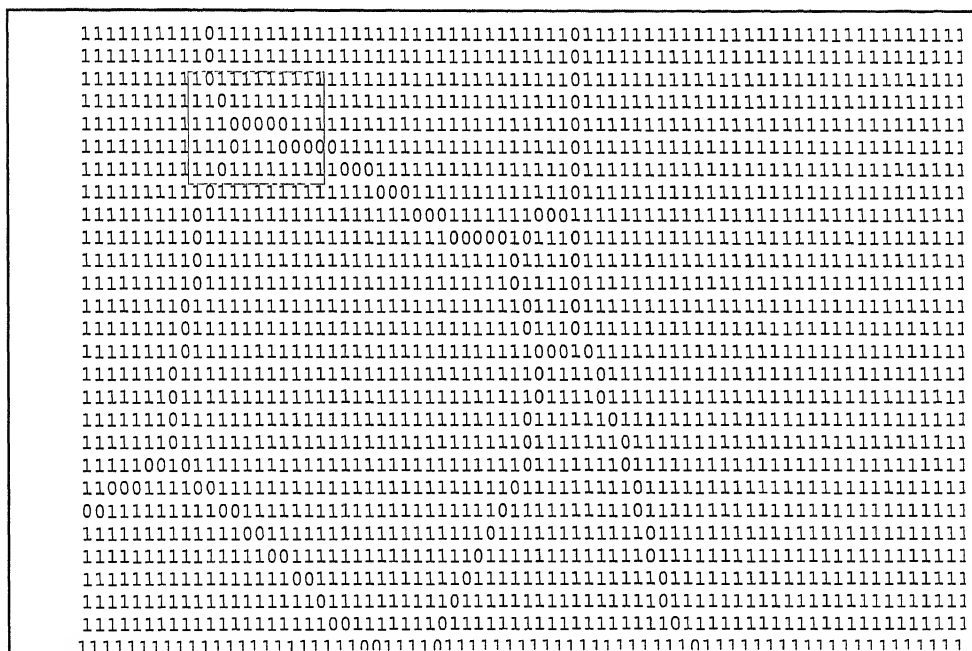


Figure 4.9: Bitmap of the marked rectangular region in figure 4.8. ‘0’ refers to grain boundary pixels and ‘1’ refers to matrix pixels.

4.2 Image Analysis Algorithm: Evaluation of Plane Dihedral Angles

4.2.1 Triple Point Detection

Refer to the bitmap of the rectangular section of the image (figure 4.8) of microstructure as seen in fig 4.9. Each grain boundary pixel (pixel value of '0') was considered at the center of a 3x3 matrix. If a grain boundary pixel had exactly three other boundary pixels in its neighbourhood then the pixel was marked as a possible triple point and subjected to further evaluation by comparing it with a pre-generated list of valid triple point configurations. A partial list of valid configurations in the form of 3x3 matrices are shown in figure 4.10. The remaining configurations can be generated by rotating the individual matrices through $\frac{\pi}{2}, \pi$ and $\frac{3\pi}{2}$ thereby generating a total of 12 valid configurations.

However there are many other configurations possible in which the central '0' pixel has three neighbouring pixels, but are not included in the above list. Some examples are listed in figure 4.11. The reason for this becomes evident from figure 4.11, which shows a magnified bitmap view of the rectangular region marked in figure 4.9. The (row, column) coordinates of each pixel in the bitmap can be referred relative to the left-top pixel, whose coordinate is assigned (1,1), as shown in figure 4.11. On examining this figure, it is clear that grain boundary pixels at coordinates (3,6), (3,7), (4,7) and (4,8) are not triple points even though each of these pixels have exactly three neighboring grain boundary pixels, such triple points have been termed as pseudo triple points. Similarly the grain boundary pixels at coordinates (3,4) and (4,3) are also pseudo triple points. However the triple point at coordinate (3,3) is an actual triple point, which matches with the configuration shown in figure 4.10a rotated through π . Thus,

the pseudo triple points can be classified according to the following general rule:
If any two of the three neighboring pixels share a boundary among themselves, which means that if any two neighbouring pixels have same 'x' coordinate and the 'y' coordinate differs by one and vice-versa then the triple point is classified as a pseudo triple point This rule was found to be appropriate by comparing the results of the automated detection of triple points with manual detection.

1	0	1
0	0	1
1	1	0

(a)

1	1	0
0	0	1
1	1	0

(b)

1	1	0
1	0	1
0	1	0

(c)

Figure 4.10. A partial configurations of valid triple points. Remaining configurations can be generated rotating the matrices through $\pi/2$, π and $3\pi/2$

	1	2	3	4	5	6	7	8	9	10
1	0	1	1	1	1	1	1	1	1	1
2	1	0	1	1	1	1	1	1	1	1
3	1	1	0	0	0	0	0	1	1	1
4	1	1	0	1	1	1	0	0	0	1
5	1	0	1	1	1	1	1	1	1	0
6	0	1	1	1	1	1	1	1	1	1

Figure 4.11: Illustration of pseudo triple points (refers to rectangular region marked in figure 4.9)

4.2.2 Evaluation of Plane Dihedral Angles (PDA)

Starting from each of the detected triple points, as described in section 4.2.1, the coordinates of the pixels of the grain boundary segments were traced till the segment terminated into another triple point or till no further touching pixels could be found. The algorithm developed for the same is at appendix A.1. At this stage, if a particular segment was found to be very short (less than four pixel length) then the associated triple point was considered as a pseudo-triple point and therefore eliminated from the analysis. Manual examination of the microstructure confirmed the validity of this procedure. The output of the program is at appendix B.1.

A least squares procedure was employed to fit polynomials on the pixels of the grain boundary segments at each triple point. The form of the polynomial chosen for the purpose is given as:

$$y = y_0 + a_1(x - x_0) + a_2(x - x_0)^2 + \dots + a_m(x - x_0)^m \quad (4.2.2.1)$$

Where 'm' is the polynomial order, a_1, a_2, \dots, a_m are the coefficients. This form of the polynomial was modified to ensure that the fitted curve is constrained to pass through the triple point.

A least squares procedure was adopted to determine the values of the polynomial coefficients. This involved minimization of the sum of squares (R^2) of the following equation:

$$R^2 = \sum_{i=1}^n [y_i - y]^2 \quad (4.2.2.2)$$

where, y_i is the measured value and y is given by equation (1) for the i th pixel. Derivative of equation 4.2.2.2 with respect to each of the unknown coefficients of the polynomial is performed and equated to zero to develop a set of equations which was solved by gauss-elimination technique.

Each grain boundary segment was fitted with polynomial of increasing degree of order, starting with the order of 1. In order to select the best polynomial order, a ratio F was evaluated to compare the goodness-of-fit (quantified by sum of squares, R^2) of two polynomials of orders m and $m+1$

$$F = \frac{(R_1^2 - R_2^2) / R_1^2}{(\gamma_1 - \gamma_2) / \gamma_1} \quad (4.2.2.3a)$$

and

$$\gamma_1 = n - (m + 1) \quad \text{and} \quad \gamma_2 = n - (m + 2) \quad (4.2.2.3b)$$

where, γ = degree of freedom, n = number of data points and the subscripts 1 and 2 refer to the polynomials of orders m and $m+1$ respectively.

If the fit of the higher order polynomial (order = $m+1$) has a larger sum of squares, i.e., $R_2^2 > R_1^2$, then clearly the fit becomes worse on increasing the order and therefore, the lower order polynomial (order = m) is chosen as the best order. In terms of the F -ratio (equation 4.2.2.3a), the value of F will be negative. However, in general with increasing polynomial order the sum of squares value is expected to reduce because the higher order equation has more adjustable parameters. Thus, a further analysis of the F -ratio was done.

In going from lower order to the higher order polynomial, if the relative decrease in the sum-of-squares was less than the relative decrease in the degree of freedom (i.e., $F \leq 1$), then the polynomial with the lower order was selected as the best polynomial order. On the other hand, if the value of F was greater than 1, i.e., the relative decrease in the sum of squares was larger than the relative decrease in the degree of freedom, then the higher polynomial order was accepted and the above analysis was repeated by further incrementing the polynomial order.

The correlation coefficient (Chapra 1989), r , for each polynomial fit was determined by using the following equation:

$$r^2 = \frac{S^2 - R^2}{S^2} \quad (4.2.2.4)$$

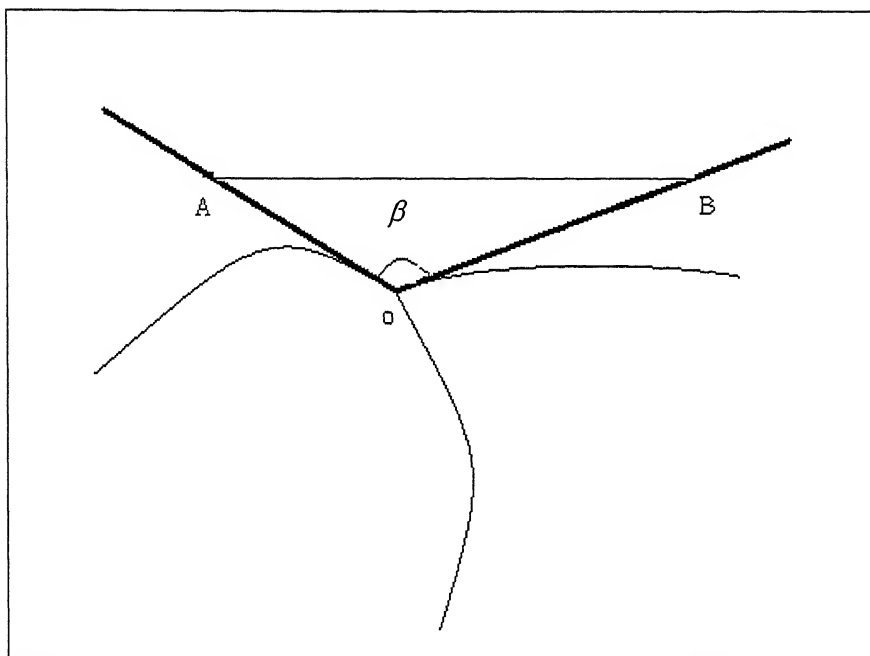
$$\text{where, } S^2 = \sum_{i=1}^n (y_i - \bar{y})^2 .$$

For a perfect fit $R^2 = 0$ and $r^2 = 1$, signifying that the polynomial fit truly fits all the coordinates of each grain boundary. While r^2 close to zero signifies a poor fit. The correlation coefficient was used to decide whether to accept or reject a particular fit, as discussed in the section (**chapter 5 results and discussions**).

The plane dihedral angles were determined as the angles between the tangents to the grain boundary segments at the triple points. Figure 4.12 illustrates a typical example of three tangent lines (calculated by differentiating the fitted curves) at a triple point. Plane dihedral angle, β , is given by

$$\cos \beta = \frac{(OA)^2 + (OB)^2 - (AB)^2}{2(OA)(OB)} \quad (4.2.2.5)$$

Similarly, the other two plane dihedral angles can also be determined.



4

Figure 4.12: Tangents (thick lines) drawn to the fitted grain boundary segments (thin lines) at a triple point 'O'. 'A' and 'B' are arbitrary points on the tangents.

4.3 3-Dimensional Stereological Probe: The Disector

4.3.1 Transformation of Plane Dihedral Angles (PDA) to True Dihedral Angles (TDA)

Figure 4.13 shows the geometry of a disector cutting the grain boundary planes meeting at a triple edge. The two sections, S_1 and S_2 (with a separating distance of h) are parallel to the x-y plane. The vectors \hat{n}_1, \hat{n}_2 and \hat{n}_3 are the unit normals to the grain boundary planes 1, 2 and 3 respectively. The orientation of the triple edge is defined by the vector \vec{E} . The intersection of the disector with the grain boundary planes result in linear grain boundary segments meeting at triple points 'Q' and 'P' on the parallel sections S_1 and S_2 respectively. The orientation of the grain boundary segments PA, PB and PC are defined by the unit vectors \hat{r}_1, \hat{r}_2 and \hat{r}_3 respectively. The plane dihedral angles, β_1, β_2 and β_3 , are the angles between the vectors \hat{r}_1 and \hat{r}_2 , \hat{r}_2 and \hat{r}_3 , and \hat{r}_3 and \hat{r}_1 respectively. The coordinates of the triple points 'Q' and 'P' are denoted as (t_{x1}, t_{y1}) and (t_{x2}, t_{y2}) respectively.

The true dihedral angles α_1, α_2 and α_3 are given by the following general equations (refer to figure 4.13):

$$\cos \alpha_1 = \hat{n}_1 \cdot \hat{n}_2 = \frac{\vec{E} \times \hat{r}_1}{|\vec{E} \times \hat{r}_1|} \cdot \frac{\vec{E} \times \hat{r}_2}{|\vec{E} \times \hat{r}_2|} \quad (4.3.1.2a)$$

$$\cos \alpha_2 = \hat{n}_2 \cdot \hat{n}_3 = \frac{\vec{E} \times \hat{r}_2}{|\vec{E} \times \hat{r}_2|} \cdot \frac{\vec{E} \times \hat{r}_3}{|\vec{E} \times \hat{r}_3|} \quad (4.3.1.2b)$$

$$\alpha_3 = 2\pi - (\alpha_1 + \alpha_2) \quad (4.3.1.2c)$$

The following transformation of coordinates can be applied without any loss of generality (see figure 4.13). The origin is translated from point O to the triple point P followed by a rotation about the z-axis (angle between the grain boundary segment PA and the x-axis), through ϕ which is given by

$$\cos \phi = \frac{(x_1 - t_{x2})}{[(x_1 - t_{x2})^2 + (y_1 - t_{y2})^2]^{1/2}} \quad (4.3.1.3a)$$

or

$$\sin \phi = \frac{(y_1 - t_{y2})}{[(x_1 - t_{x2})^2 + (y_1 - t_{y2})^2]^{1/2}} \quad (4.3.1.3b)$$

where, x_1 and y_1 are the coordinates of an arbitrary point on the segment PA. The above transformation results in the unit vector \hat{r}_1 coinciding with the transformed x' -axis. In this transformed system of axes (x', y', z') the coordinates of the triple point P is (0,0,0) and the coordinates of triple point Q is (t_x, t_y, h):

$$t_x = (t_{x1} - t_{x2}) \cos \phi + (t_{y1} - t_{y2}) \sin \phi \quad (4.3.1.4a)$$

$$t_y = (t_{y1} - t_{y2}) \cos \phi - (t_{x1} - t_{x2}) \sin \phi \quad (4.3.1.4b)$$

Thus, from figure 4.13, the vector \vec{E} and the unit vectors \hat{r}_1 and \hat{r}_2 are given by the following expressions:

$$\vec{E} = \hat{i} t_x + \hat{j} t_y + \hat{k} h \quad (4.3.1.5a)$$

$$\vec{r}_1 = \hat{i} \quad (4.3.1.5b)$$

$$\vec{r}_2 = \hat{i} \cos \beta_1 + \hat{j} \sin \beta_1 \quad (4.3.1.5c)$$

The true dihedral angle α_1 can be obtained by substituting equations (4.3.1.5) in equation (4.3.1.2a):

$$\cos \alpha_1 = \frac{\hat{j}h - \hat{k}t_y}{(h^2 + t_y^2)^{1/2}} \cdot \frac{-\hat{i}h \sin \beta_1 + \hat{j}h \cos \beta_1 + \hat{k}(t_x \sin \beta_1 - t_y \cos \beta_1)}{[h^2 + (t_x \sin \beta_1 - t_y \cos \beta_1)^2]^{1/2}}$$

On simplification :

$$\cos \alpha_1 = \frac{h^2 \cos \beta_1 - t_y (t_x \sin \beta_1 - t_y \cos \beta_1)}{(h^2 + t_y^2)^{1/2} [h^2 + (t_x \sin \beta_1 - t_y \cos \beta_1)^2]^{1/2}} \quad (4.3.1.6)$$

Similarly equations for α_2 can be derived by performing a transformation of coordinates in which the rotation about the z-axis is done through $\phi + \beta_1$, (instead of ϕ , as above)

The expression for α_2 (which has the same form as equation 4.3.1.6) is given below:

$$\cos \alpha_2 = \frac{h^2 \cos \beta_2 - t'_y (t'_x \sin \beta_1 - t'_y \cos \beta_2)}{(h^2 + t'^2_y)^{1/2} [h^2 + (t'_x \sin \beta_2 - t'_y \cos \beta_2)^2]^{1/2}} \quad (4.3.1.7)$$

where t'_x and t'_y (similar to t_x and t_y in equations 4.3.1.4) are:

$$t'_x = (t_{x1} - t_{x2}) \cos(\phi + \beta_1) + (t_{y1} - t_{y2}) \sin(\phi + \beta_1) \quad (4.3.1.8a)$$

$$t'_y = (t_{y1} - t_{y2}) \cos(\phi + \beta_1) - (t_{x1} - t_{x2}) \sin(\phi + \beta_1) \quad (4.3.1.8b)$$

Above derived disector technique has been used to estimate true dihedral angles from digital images, as discussed below.

4.3.2 Computation Methodology of True Dihedral Angle

Figure 4.14a shows a typical unprocessed grayscale image of a polycrystalline microstructure of 316L austenitic stainless steel annealed at 900° C for 60 minutes. This image is the first disector section S_1 as already explained in figure 3.1. The figure also shows the micro hardness indents which were used as reference markers. The grayscale images were binarized, etching/polishing artifacts were removed, and

the grain boundaries were thinned and pruned by an image-processing scheme developed in section 4.1. The result of the image-processing scheme on the image in figure 4.14a is shown in figure 4.14b. The sample was re-polished and re-etched to generate the disector section S_2 , as shown in figures 4.14c and 4.14d (processed image). The size of the indents in sections S_1 and S_2 (see the unprocessed images of the two sections in figure 4.14a and 4.14c respectively) were substituted in equation 3.1 to determine the disector thickness, h . For accurate measurement of the disector thickness the indent size in section S_2 was measured after polishing but prior to etching. This avoided any errors resulting due to the distortion of the indent due to etching. The various disector thicknesses obtained for samples subjected to different annealing treatments are shown in table 3.2.

In order to match the center of the corresponding indents in the two images of sections S_1 and S_2 with each other precisely, an annotation mark was made at the center of the indents. The images of the two sections were merged such that the centers of the indents were matched. Care was taken to mount the sample in such a way that almost no rotation was needed to match the center of the indents. The result of the match of the first and the second image was found to be within 2 to 3 pixel accuracy, as shown in the composite image in figure 4.14e. The light image is from section S_1 and the dark image is from section S_2 in figure 4.14e. Triple point pairs similar to the triple point pair P and Q shown in figure 4.13 is shown in figure 4.14f which is enlarged image of the rectangular section marked in figure 4.14e. For example the pair of triple point marked as '1' (belongs to S_1) and '2' (belongs to S_2) are from a single triple edge. The varying relative shift (in the 2-dimensional plane) of triple points in section S_2 , with reference to these on section S_1 is done to the random orientation of the triple edges. It may also be noted that there are many triple points in section S_1 (see rectangular region in figure 4.14f), which do not have any corresponding point in section S_2 . This is because of improper etching in section S_2 it could also arise if the disector thickness is larger than the size of the grains.

The measurement of the coordinates of triple points orientation of the grain boundary segments, and the plane dihedral angles in the binarized images of plane sections S_1 and S_2 were performed by an automated image analysis algorithm described in section 4.2. Corresponding triple point pairs in the two sections were identified by an

algorithm, which used the displacement of triple points in section S2 relative to these in section S1, and the closest neighbors were paired (program in appendix A.5). This simple algorithm was successful in microstructural images containing large grains (such as the structure obtained in samples annealed at temperatures of 1000° C and above). For microstructures with smaller grains the procedure was only partially successful and manual intervention was required. This problem could be eliminated by additionally comparing the orientation of the corresponding grain boundary segments. However, no further attempts were made in this direction. Finally, equations 4.3.1.6, 4.3.1.7 and 4.3.1.2a were used to estimate the true dihedral angles and their distributions.

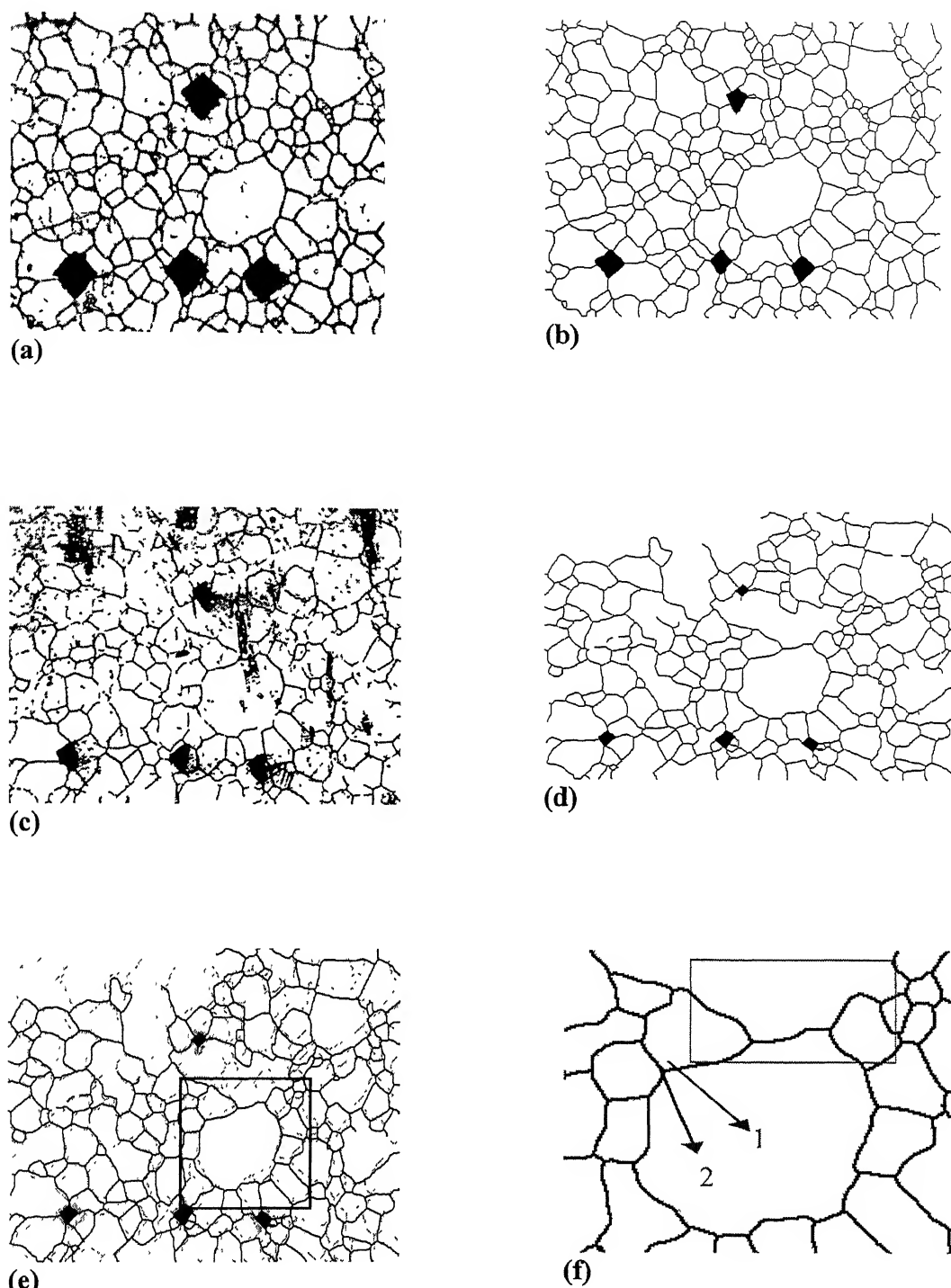


Figure 4.14. Microstructures of 316L austenitic stainless steel with microhardness indents as reference markers: (a) unprocessed image of disector section S_1 ; (b) Processed image of disector section S_1 ; (c) Unprocessed image of disector section S_2 ; notice the reduction in the sizes of the indents; (d) Processed image of section S_2 . (e) Merged sections S_1 and S_2 using centers of indents as reference; (f) image of marked section in figure 2e indicating a random triple edge.

CHAPTER 5

RESULTS AND DISCUSSIONS

As mentioned in chapter 3, the commercially available 316L austenitic stainless steel was obtained and were cold rolled to reduce thickness by 92% in more than 25 successive passes. To achieve range of grain sizes the cold rolled samples were annealed at different temperatures and times as indicated in table 5.1. The microstructure of the samples was characterized after capturing digital images and processing them as shown in figure 5.1. In general the digital images were converted into bitmaps and was used as input to the computer algorithms developed to evaluate dihedral angles. The programs are listed in appendix A and the outputs of these programs are illustrated in appendix B.

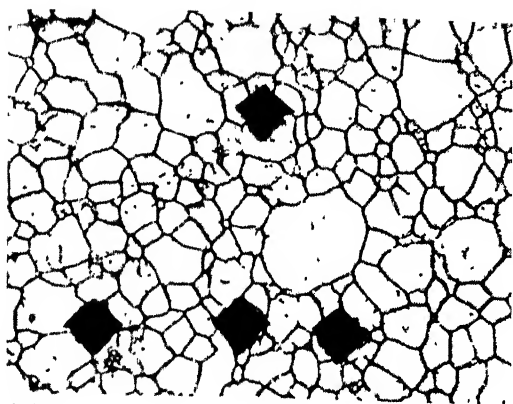
In order to ensure measurement accuracy of plane dihedral angles by the automated technique, the correlation coefficients of the best-fit polynomials were checked. Figure 5.2, shows typical example of the polynomial fit produced along with the correlation coefficients at a triple point. A statistical sample of manual measurements was compared with automated measurements along with the correlation coefficients (figure 5.2). By such comparisons it was observed that in the case of each large discrepancy between manual and automated measurement the correlation coefficient for one or more of the best-fit polynomial was less than 0.40. Thus for all subsequent automated measurements of plane dihedral angles, if the correlation coefficient was less than 0.40 then the associated triple point was rejected from the analysis. In the analysis of the microstructures of the samples of 316L austenitic stainless steel such a procedure resulted in approximately 5% rejection of the triple points. Thus by using the above technique, it was possible to obtain a statistically significant sample of the plane dihedral angles. Table 1 gives the total number of triple points identified and the valid triple points finally evaluated per image.

Considering all the measured plane dihedral angles, histograms of the relative frequencies of plane dihedral angles were generated. Figure 5.3 shows a histogram of the relative frequencies of plane dihedral angles for the sample annealed at various temperatures. Figure 5.4 shows the variation of standard deviations of plane dihedral angles as a function of annealing treatment. From this figure it is evident that with increasing annealing temperature, the spread of the distribution reduces about the mean value of 120° . This is expected because with increasing annealing temperature grain boundaries tend to equilibrate [4].

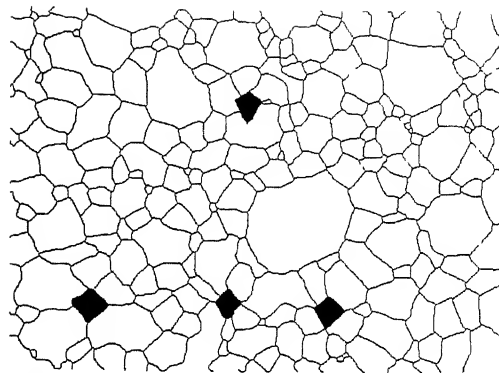
Table 5.1: Statistical parameters of plane dihedral angles

NN EMP. 'C)	ANN TIME (Minutes)	GRAIN SIZE* (μm)	STATISTICAL PARAMETERS					
			NO OF TRIPLE POINTS EVALUATED	STANDARD DEVIATION (PDA)	DISECTOR THICKNESS (μm)	STANDARD DEV (TDA)	STD. DEV (GBE)	MEAN (GBE)
900	60	24.916	199	26.0393	3.80	26.95	0.186	0.57
900	180	26.69	160	26.508	4.50	24.64	0.187	0.5708
950	60	35.93	110	24.89	3.18	23.38	0.151	0.62
1000	60	53.39	120	24.72	6.16	18.79	0.14	0.70
1050	60	62.29	52	23.2	6.25	18.153	0.139	0.744

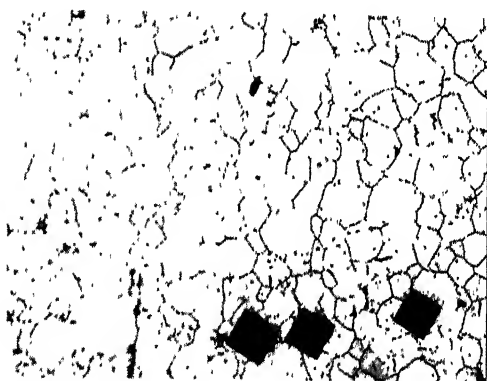
* mean intercept length has been taken as a measure of grain size



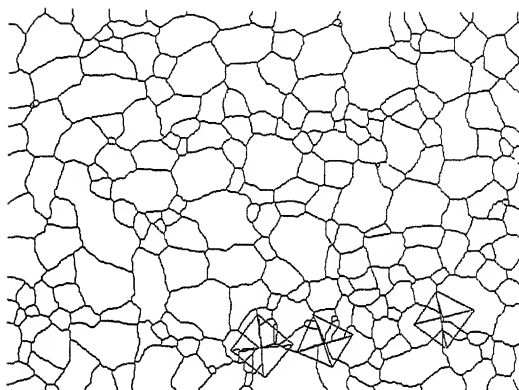
(a)



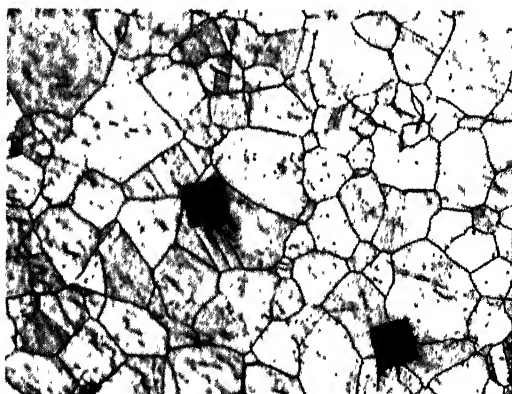
(b)



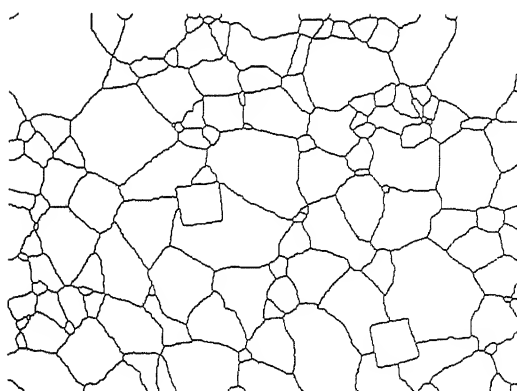
(c)



(d)



(e)



(f)

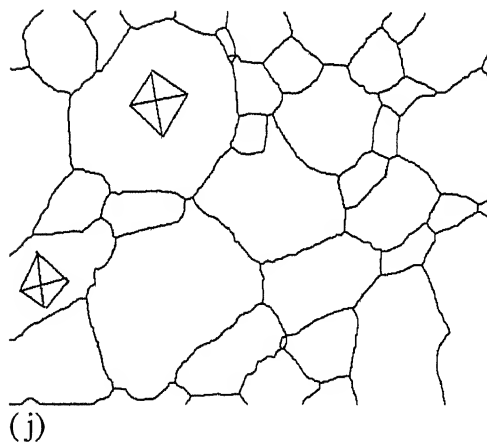
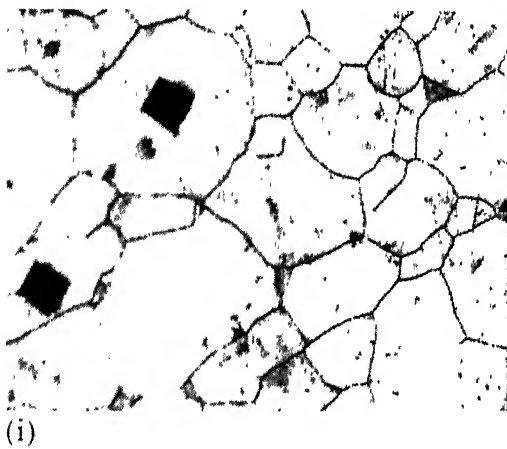
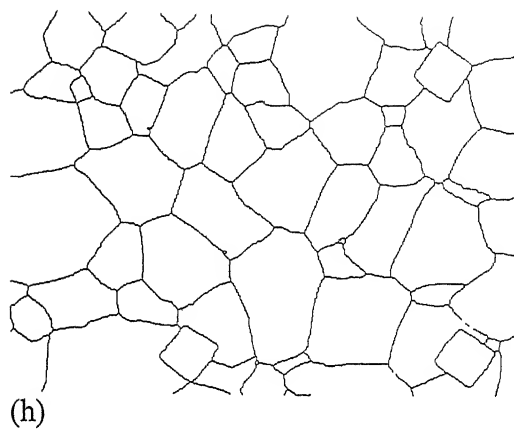
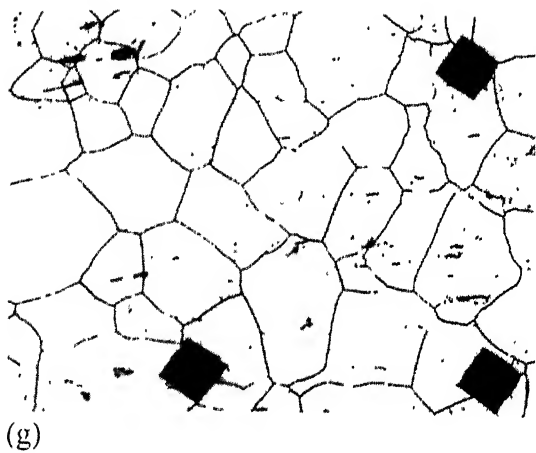


Figure 5.1: Unprocessed and processed image annealed at (a) and (b) 900°C for 60 min; (c) and (d) 900°C for 180 min; (e) and (f) 950°C for 60 min; (g) and (h) 1000°C for 60 min; (i) and (j) 1050°C at 60 min.

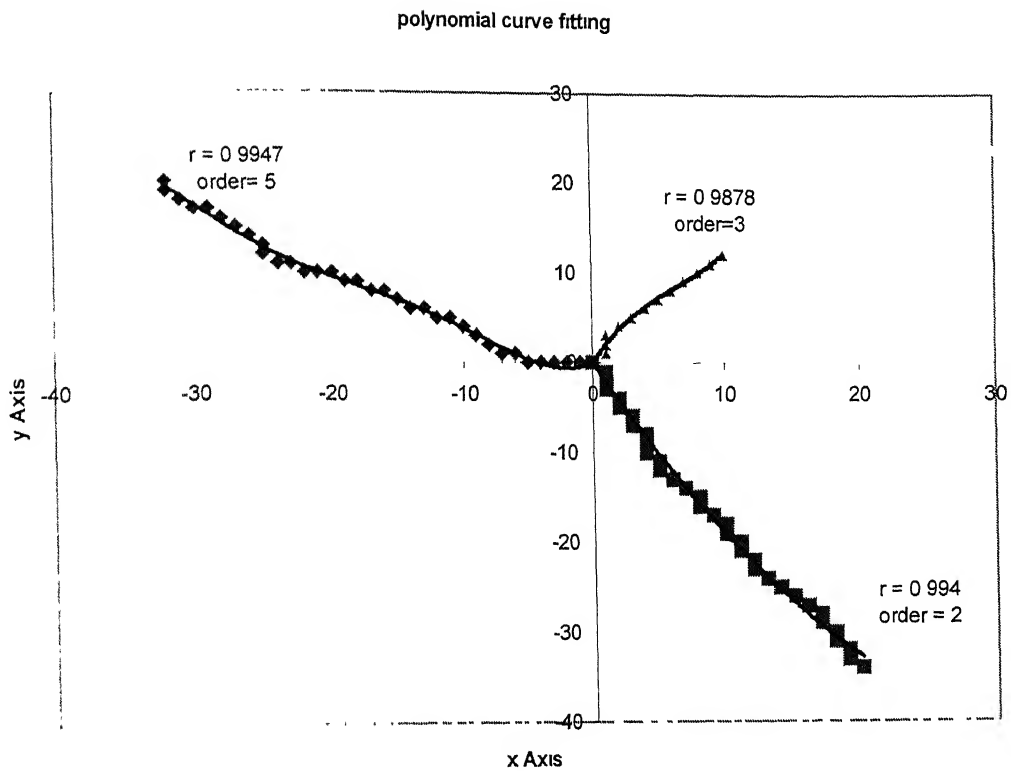
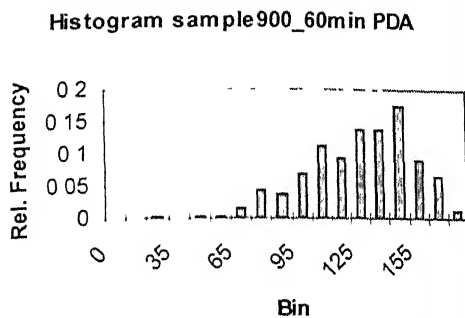
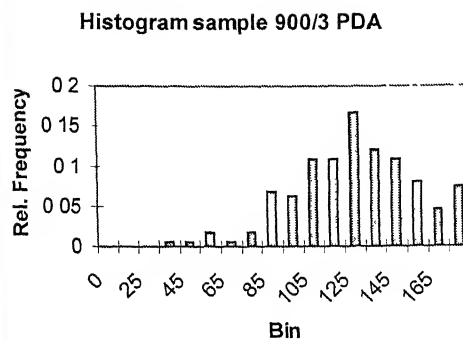


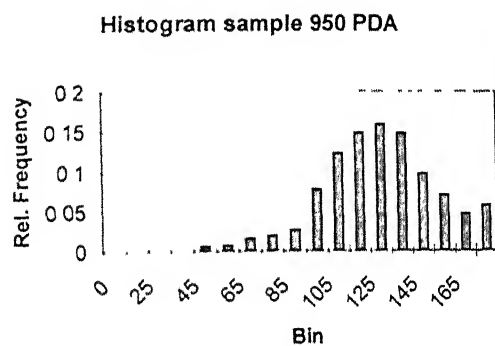
Figure 5.2: Typical example of polynomial fits on grain boundary segments at a triple point



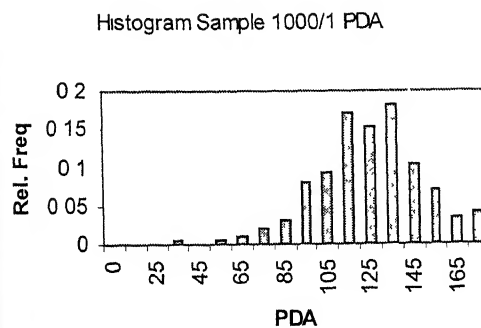
(a)



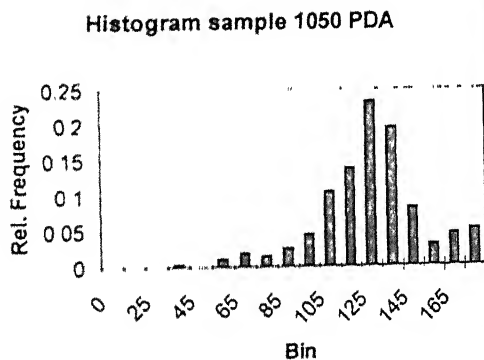
(b)



(c)



(d)



(e)

Figure 5.3: PDA distribution of sample annealed at (a) 900°C for 60 minutes; (b) 900°C for 180 minutes; (c) 950°C for 60 minutes; (d) 1000°C for 60 minutes; (e) 1050°C for 60 minutes.

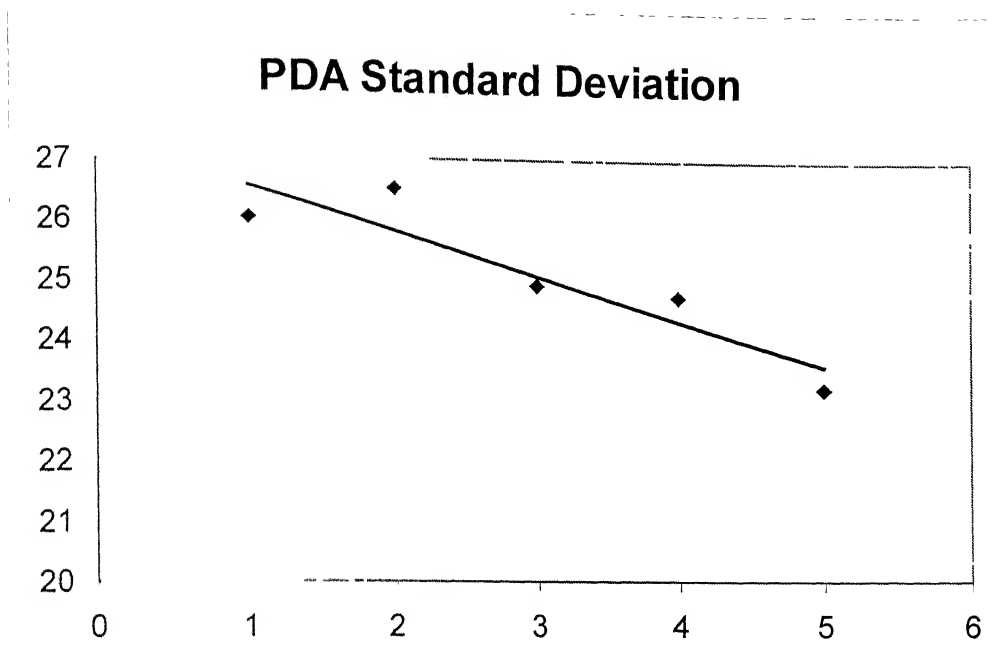


Figure 5.4: Illustration of standard deviation of plane dihedral angles as a function of annealing temperatures

As discussed in section 4.3 the two disector sections S_1 and S_2 were processed as mentioned in section 4.2 to obtain plane dihedral angles. Also, Corresponding triple point pairs in the two sections were identified by an algorithm (appendix A.5), which used the displacement of triple points in section S_2 relative to those in section S_1 , and the closest neighbors were paired to form a grain boundary edge as shown in figure 4.13. Equations 4.3.1.6, 4.3.1.7 and 4.3.1.2 were used to estimate the true dihedral angles and their distributions. Histograms of the relative frequencies of the true dihedral angles were generated, which is shown in figure 5.5. Figure 5.6 shows the variation in standard deviations as a function of annealing treatment. From the figures it was evident that as the annealing temperatures were increased the standard deviations decreased and the peaks shifted towards 120° class. This explains that at higher temperatures the grain boundaries tend to equilibrate as also mentioned in the case of plane dihedral angles. Thus in the fine grain microstructures the standard deviations of TDA is higher and the distribution peak is away from 120° class in comparison to coarser grains microstructure. This indicates that in the fine grain structures there is a state of non-equilibrium at the grain boundaries.

On comparing the standard deviations of the plane dihedral angles with corresponding standard deviations of true dihedral angles it was found that in general (particularly samples annealed at 1000° and 1050° C) the standard deviation of TDA was lower than corresponding standard deviation of PDA. This can be understood in terms of the factors contributing to the spread of the distribution. The standard deviation (or variance) of the plane dihedral angle distribution is affected by the variation in the true dihedral angles and the variation due to sampling of randomly oriented triple edges by the sectioning plane (see Figure 2.4b). The latter contribution is eliminated during the process of transformation from plane to true dihedral angles leading to a relatively smaller spread in the distribution. This fact would become significant in studies attempting to relate the factors, such as, deformation temperature, strain, etc. to the distribution of dihedral angles.

The samples annealed at 900° C, the standard deviation of TDA is comparable to that of PDA. Examination of the microstructural data in these samples revealed significant

difference (as much as 20%) between plane dihedral angles in the two disector planes (S_1 and S_2) at same triple edges. This suggests that some of the triple edges must be curved (while the analysis developed here requires a straight edge)

In order to obtain the distribution of relative grain boundary energies from distribution of TDA, the following expression was used:

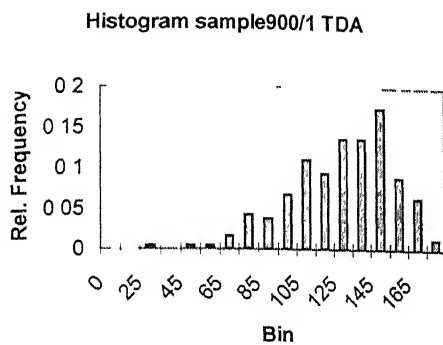
$$\frac{\gamma_1}{\sin \alpha_1} = \frac{\gamma_2}{\sin \alpha_2} = \frac{\gamma_3}{\sin \alpha_3} \quad (5.1)$$

where γ_1 , γ_2 and γ_3 the energies of the grain boundary planes making a triple edge. Using equation 5.1, the ratio of the minimum energy to maximum energy:

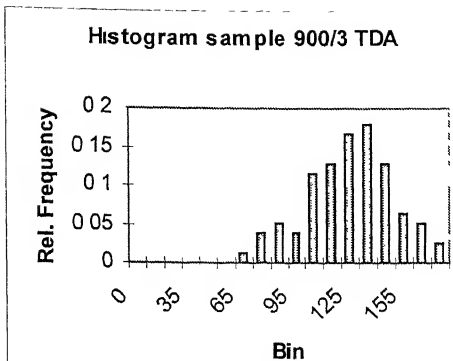
$$\frac{\gamma_3}{\gamma_1} = \frac{\sin \alpha_1}{\sin \alpha_3} \quad (5.2)$$

where $\gamma_1 \geq \gamma_2 \geq \gamma_3$.

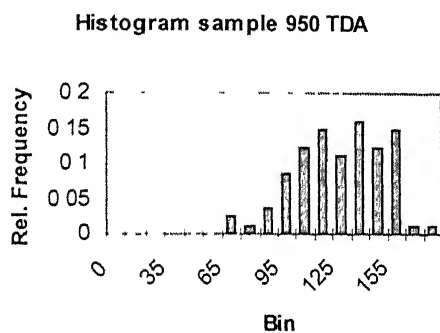
Figure 5.8 shows the distribution of the relative grain boundary energy ($\frac{\gamma_3}{\gamma_1}$) and the statistical parameter of the distribution are shown in table 5.1 (see figure 5.7). It could be seen that the grain boundary energy ratio increases towards unity with increasing annealing temperatures, which is expected, as grain boundary energies tend to equilibrate at higher temperatures.



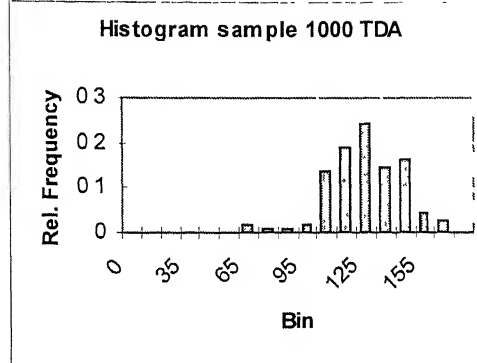
(a)



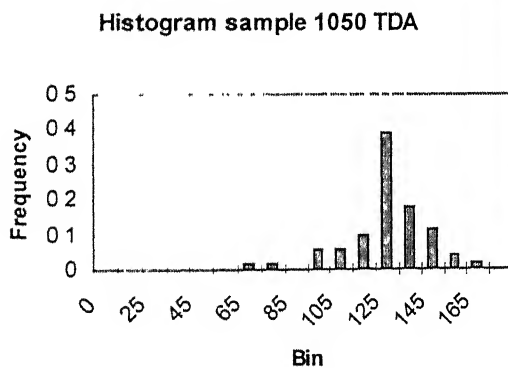
(b)



(c)



(d)



(e)

Figure 5.5: TDA distribution of sample annealed at (a) 900°C for 60 minutes; (b) 900°C for 180 minutes; (c) 950°C for 60 minutes; (d) 1000°C for 60 minutes; (e) 1050°C for 60 minutes

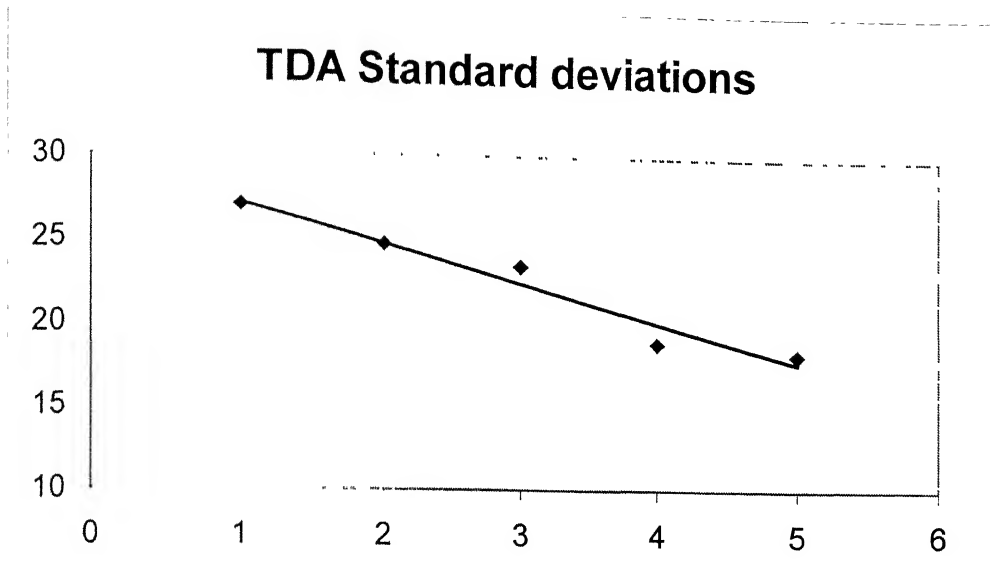


Figure 5.6: Illustration of standard deviation of true dihedral angles as a function of annealing temperatures.

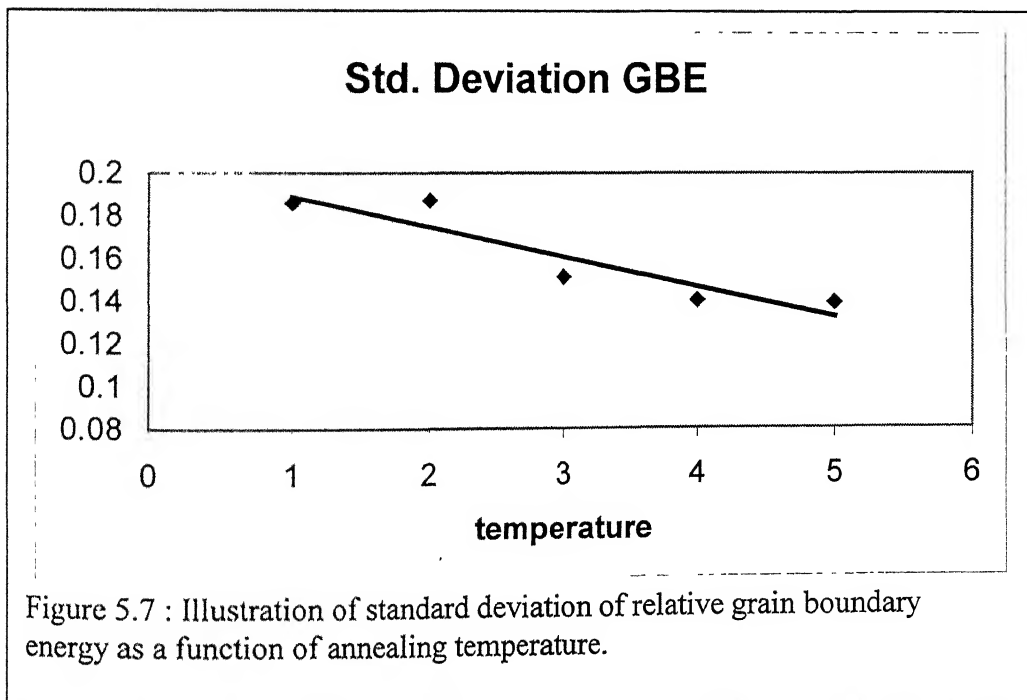
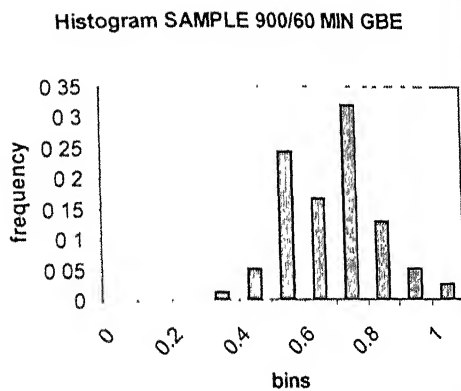
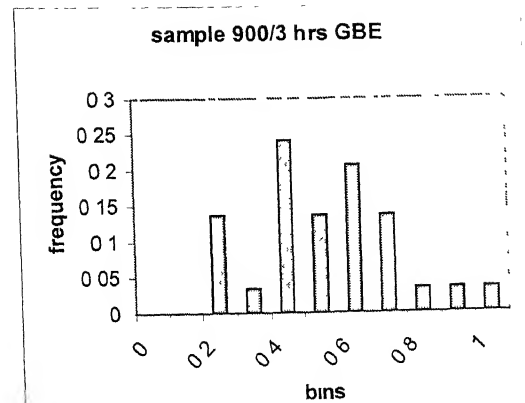


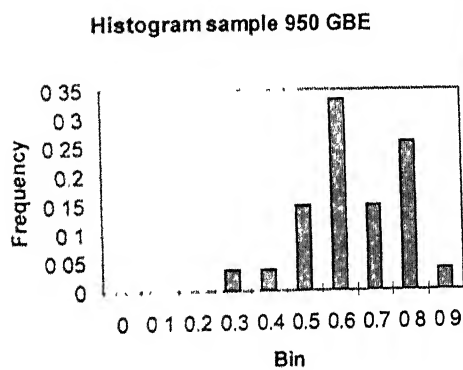
Figure 5.7 : Illustration of standard deviation of relative grain boundary energy as a function of annealing temperature.



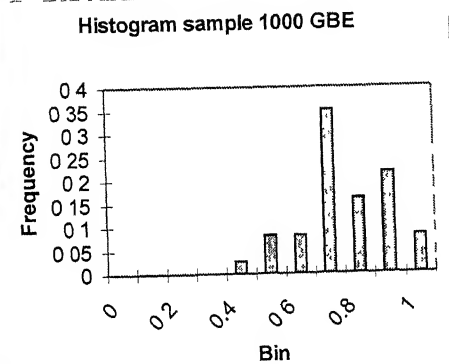
(a)



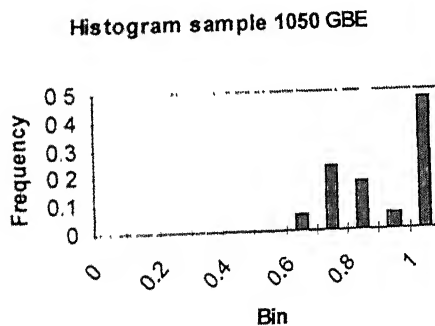
(b)



(c)



(d)



(e)

Figure 5.8: Relative grain boundary energy distribution of sample annealed at (a) 900°C for 60 minutes; (b) 900°C for 180 minutes; (c) 950°C for 60 minutes; (d) 1000°C for 60 minutes; (e) 1050°C for 60 minutes

CHAPTER 6

CONCLUSIONS

The major conclusions of this study can be summarized as follows:

1. An automated methodology was evolved integrating image acquisition, processing and analysis to determine plane dihedral angles in polycrystalline materials. The algorithm developed uses morphological operations to clean microstructural images and performs thinning and pruning of grain boundaries. The automated procedure identifies triple points and the pixel coordinates of the corresponding grain boundary segments. The dihedral angles at each triple point are calculated by selecting the best-fit polynomials on the grain boundary segments on a sample of polycrystalline austenitic stainless steel. Parameters such as standard deviation of plane dihedral angles are compared as a function of annealing temperature. The results so obtained corroborated the efficacy of the scheme developed for evaluating dihedral angles.
2. The standard deviations of plane dihedral angles decreases with increasing annealing temperatures, that is, the spread of the plane dihedral angle distribution reduces about the mean value of 120° class. This is expected because with increasing annealing temperatures grain boundaries tend to equilibrate.
3. It is also evident that as the annealing temperatures were increased the standard deviations of true dihedral angles decreased and the peaks shifted

towards 120° . This explains that at higher temperatures the grain boundaries tend to equilibrate as also mentioned in the case of plane dihedral angles.

4. The spread of the transformed true dihedral angles distribution is considerably less than the spread of the plane dihedral angles distribution. This has been attributed to the fact that the transformation from plane to true dihedral angles distribution results in the elimination of the variance due to random sampling of triple edges by the plane of polish. It has been shown that the parameters, such as, standard deviation and peak frequency of true dihedral angles distribution are much more sensitive to deformation as compared to the parameters for plane dihedral angles distribution.
5. The standard deviations of the ratio of grain boundary energy decreased with increasing annealing temperatures and the mean value of the ratio progressively increased (values moved closer to 1.0), as expected.
6. The technique developed in this work to estimate the distribution of true dihedral angles from the measured PDA is not just limited to 316L austenitic stainless steel but can also be applied to any polycrystalline material.

The suggestions and the scope for the future research are given below

- To match the corresponding triple points to form a grain boundary edge the algorithm in appendix A.5 is used, the procedure was only partially successful and manual intervention was required for lower temperature annealed samples. This problem can be resolved by additionally comparing the orientation of corresponding grain boundary boundaries.
- The study can be applied in the field of textures and sintered microstructures.

- To further increase the statistical accuracy of the disector technique the number of triple point analyzed should be increased to several hundred. This can be achieved by creating a montage consisting of perfectly matched adjacent images from a plane section.

References

1. Adams, B.L; Kinderlehrer, D.; Mullins, W.W; Rollet, A.D; Ta'asan, S.: "Extracting relative grain boundary energy and mobility functions from the geometry of microstructures", *Scripta Materialia*, vol 38, 1998.
2. B.L. Adams, D. Kinderlehrer, I. Livshits, D. E. Mason, W.W. Mullins, G.S. Rohrer, A.D. Rollett, D.M. Saylor, S. Ta'asan, Chun-te Wu, Extracting Grain Boundary and Surface Energy from Measurement of Triple Junction Geometry," *Interface Science*, vol 7, (1999) 321-337.
3. Chapra, S.C., Canale, R.P.: McGraw Hill, (1989).
4. DeHoff, R.T. (1986) estimation of dihedral angles from stereological counting measurements, *Metallography* 19:209-217.
5. DeHoff and Rhines: *Quantitative Microscopy*, McGraw Hill (1968).
6. Gundersen, H.J.G. (1977) Note on estimation of numerical density of arbitrary profiles: The edge effect, *J.Microsc.* 111. 219-223.
7. Jain, A.K.: "Fundamentals of digital image processing", prentice hall, (2000)
8. Murr, L.E.: *Journal of Applied Physics*, 39:5557, 1968.
9. Murr, L.E.; Smith, P.J, Glimore, C.M.: *Phil. Mag* (a), 17:89,1968.
10. Murr, L.E.: *Phys Stat. Sol.* (a), 1:487,1970.
11. S.A.Saltykov, "The determination of the size distribution of particles in an opaque materials from a measurement of the size distribution of their sections," *stereology* , edited by H.Elias, proc second int. cong. For stereology, new york, (1967) 163.
12. Sandeep Sangal and Tangri, K.: *Scripta Materialia*, vol 23, 2079(1989).
13. Serra, J.: "Image analysis and mathematical morphology", vol 1, academic press London (1982).
14. Singh, K.K.; Sangal, S.; Murthy, G.S: Effect of Grain Size and Microstructural Correlations in the Plastic Behavior of 316L Austenitic Stainless Steel at Elevated Temperatures, *Mat. Sc. & Tech.*, 2001 (accepted).
15. Singh, K.K.; Sangal, S.; Murthy, G.S: Effect of Microstructural Correlations in the Plastic Behavior of 316L Austenitic Stainless Steel at Elevated Temperatures, *Mat. Sc. & tech.*, 2001 (accepted).
16. Singh, K.K.: "Effect of grain size and microstructural correlations in plastic behavior of stainless steel", thesis, IIT Kanpur, Apr 1998.
17. Sterio, D.C. (1984) The unbiased estimation of number and sizes of arbitrary particles using disector, *J.Microsc.* 134, 127-136.
18. Underwood, E.E.: *Quantitative stereology*, Addison- Waley Pub, (1970)
19. Vander Voort, G.F: *Metallography; principle and practice*, McGraw Hill Series in Mat. Sc. And Engg., (1999).
20. Varin, R.A. and Tangri, K: *Metall. Trans.* 12A, 1859, (1981).

References

1. Adams, B.L.; Kinderlehrer, D.; Mullins, W.W.; Rollet, A.D.; Ta'asan, S.: "Extracting relative grain boundary energy and mobility functions from the geometry of microstructures", *Scripta Materialia*, vol 38, 1998.
2. B.L. Adams, D. Kinderlehrer, I. Livshits, D. E. Mason, W.W. Mullins, G.S. Rohrer, A.D. Rollet, D.M. Saylor, S. Ta'asan, Chun-te Wu, Extracting Grain Boundary and Surface Energy from Measurement of Triple Junction Geometry," *Interface Science*, vol 7, (1999) 321-337.
3. Ch'apra, S.C., Canale, R.P.: McGraw Hill, (1989).
4. DeHoff, R.T. (1986) estimation of dihedral angles from stereological counting measurements, *Metallography* 19:209-217.
5. DeHoff and Rhines: *Quantitative Microscopy*, McGraw Hill (1968).
6. Gundersen, H.J.G. (1977) Note on estimation of numerical density of arbitrary profiles: The edge effect, *J.Microsc.* 111. 219-223.
7. Jain, A.K.: "Fundamentals of digital image processing", prentice hall, (2000)
8. Murr, L.E.: *Journal of Applied Physics*, 39:5557, 1968.
9. Murr, L.E.; Smith, P.J, Glimore, C.M.: *Phil. Mag* (a), 17:89, 1968.
10. Murr, L.E.: *Phys Stat. Sol.* (a), 1:487, 1970.
11. S.A.Saltykov, "The determination of the size distribution of particles in an opaque materials from a measurement of the size distribution of their sections," *stereology*, edited by H.Elias, *proc second int. cong. For stereology*, new york, (1967) 163.
12. Sandeep Sangal and Tangri, K.: *Scripta Materialia*, vol 23, 2079(1989).
13. Serra, J.: "Image analysis and mathematical morphology", vol 1, academic press London (1982).
14. Singh, K.K.; Sangal, S.; Murthy, G.S: Effect of Grain Size and Microstructural Correlations in the Plastic Behavior of 316L Austenitic Stainless Steel at Elevated Temperatures, *Mat. Sc. & Tech.*, 2001 (accepted).
15. Singh, K.K.; Sangal, S.; Murthy, G.S: Effect of Microstructural Correlations in the Plastic Behavior of 316L Austenitic Stainless Steel at Elevated Temperatures, *Mat. Sc. & tech.*, 2001 (accepted).
16. Singh, K.K.: "Effect of grain size and microstructural correlations in plastic behavior of stainless steel", thesis, IIT Kanpur, Apr 1998.
17. Sterio, D.C. (1984) The unbiased estimation of number and sizes of arbitrary particles using disector, *J.Microsc.* 134, 127-136.
18. Underwood, E.E.: *Quantitative stereology*, Addison- Weley Pub, (1970)
19. Vander Voort, G.F: *Metallography; principle and practice*, McGraw Hill Series in Mat. Sc. And Engg., (1999).
20. Varin, R.A. and Tangri, K: *Metall. Trans.* 12A, 1859, (1981).

21. Weibel, E.R.: Stereological Methods, Academic Press,(1980).
22. Wicksell, S.D.: The Corpuscle problem, Biometrika 17, (1925).
23. Yang, C.C.; Rollet, A.D.; Mullins, W.W.: "Measuring Relative boundary energies and mobilities in an aluminum foil from triple junction geometry", Proceedings of 1 riso international symposium on material science, 2000.

Appendix A

Computer Programs

A.1 Program to read image data and convert into coordinate system

```
/* mod_exgb.c*/
#include <stdio.h>
#include <conio.h>

#define MAXPIXEL 15000 /* size for the max points analysed*/
#define MAXSEG 1000 /* maximum pixels in a gb segment */
#define GBPIXEL '0' /* value of gb pixel */

int getap(int value[][2], int nc, int n, int p[], int m);
int test_tp(int value[][2], int p[]);

main()
{
    int p[3],q[3],gb[MAXSEG],value[MAXPIXEL][2];
    int i,j,k,l,nc,count;
    char c;

    for(i=0,j=0,nc=0;(((c=getc(stdin))!=EOF) && (nc<MAXPIXEL));)

    {
        /* reading pixel coordinates */
        if(c=='\n')
        {
            /* end of line */

            /*cprintf("(%i,%i)\n",i,j);command*/
            /*to print on the monitor*/

            i=i++;
            j=0;
        }
        if(c == GBPIXEL) {
```

```

value[nc][0] = j;          /* array stores x,y position */
value[nc][1] = i;          /* of gb pixel */
nc++;
}
if ((c == '0') || (c == '1')) j++;
}
if (nc >= MAXPIXEL) cprintf("Warning: Could not read all of image
data!\n");

```

```

count = 0;          /* set triple point counter */
for (i=0; i<nc; i++)
{
    k = getap(value,nc,i,p,3);
    if (k == 3) {      /* k=3 : it is a triple point */
        if (test_tp(value,p)==1) continue;
        if (count > 0) printf("-1\n"); /* end of line marker */
        for (j=0; j<3; j++)
        {
            l=2;
            gb[0] = i;
            gb[1] = p[j];
            do {
                if ((k = getap(value,nc,gb[l-1],q,3)) == 2) {
                    if (gb[l-2] == q[0]) /* exclude the duplicate pixel */
                        gb[l] = q[1];
                    else
                        gb[l] = q[0];
                    l++;
                }
            } while ((k == 2) && (l < MAXSEG));
            for (k=0; k<l; k++) printf("%i %i ", value[gb[k]][0], value[gb[k]][1]);
            if (j < 2) printf("-1\n"); /* end of line marker */
        }
        count++;
    }
}
printf(" %d\n", count);
printf("-99\n"); /* end of file marker */
}

```

```

int getap(int value[][2], int nc, int n, int p[], int m)
{
    /* returns no. of adjacent pixels touching pixels n */
    /* on return, p[] contains the first m adjacent points */
    int i,j,k,x,y,dx,dy;

    x = value[n][0];
    y = value[n][1];
    i = n;
    j = -1;
    k = 0;
    do {
        i = i + j;
        if (i >= nc) break;      /* no more gb pixels to check */
        if (i < 0) {
            i = n;
            j = 1;
        }
        else {
            dy = value[i][1] - y;
            if ((dy < -1) || (dy > 1)) {
                if (j == -1) {
                    i = n;
                    j = 1;
                }
                else
                    break;      /* no need to check further */
            }
            else {
                dx = value[i][0] - x;
                if ((dx > -2) && (dx < 2)) {
                    if (k < m) p[k] = i; /* found a pixel "i" touching pixel "n" */
                    k++;
                }
            }
        }
    } while(1);
    return(k);
}

```

```
int test_tp(int value[][2], int p[])
```

```
{  
    int x[3],y[3];  
    int i,j;
```

```
    for(i=0;i<3;i++)
```

```
    {  
        x[i]=value[p[i]][0];  
        y[i]=value[p[i]][1];  
    }
```

```
    for(i=0;i<3;i++)
```

```
    {  
        j=i+1;  
        if(j==3) j=0;
```

```
        if(x[i]==x[j])  
        { if(y[i]==(y[j]+1)||y[i]==(y[j]-1))  
            return(1);  
        }
```

```
        if(y[i]==y[j])  
        { if(x[i]==(x[j]+1)||x[i]==(x[j]-1))  
            return(1);  
        }  
    }
```

```
    return(0);
```

```
}
```

A.2 Program for evaluating Plane dihedral angles

```
/* modtpa3.c*/
#include <errno.h>
#include <stdio.h>
#include <alloc.h>
#include <math.h>
#include <conio.h>

#include "modcfit3.c" /* to include this file given in appendix A.3*/

#define MAX_DATA 100 /* maximum number of data points per grain
                        segment*/
#define MAX_TP 400 /*maximum number of triple points per image*/

#define DISTANCE(x1,y1,x2,y2) ((x1-x2)*(x1-x2)+(y1-y2)*(y1-y2))

typedef struct {
    int order,npts; /*order= order of polynomial fit; npts=no of pts on
                    GB line */
    float x,y,r2,*a; /* r2 = correlation coefficient*/
    } polycoeff;

typedef struct {
    float tx,ty,beta[3]; /* tx, ty = coordinates of triple points; beta= array
                          of PDA*/
    polycoeff g[3];
    } triplepoint;

int tpfit(triplepoint t[], int ntp); /* declaring five functions*/
int gb_line(float x[], float y[], int *ndat);
void pda(triplepoint t[], int ntp);
get_distance(double d[], double s1, double s2, triplepoint t, int i);
double slope(float a[], int order, int x);

main()
{
```

```

int ntp;

triplepoint t[MAX_TP];
cprintf("\nProgram Begin\n");
/* tpfit("d:\\tc\\source\\dataout.txt",t,&ntp);*/
tpfit(t,&ntp);
pda(t,ntp);
cprintf("ntp = %i\n",ntp); /* prints number of triple points
                             evaluated*/
}

void pda(triplepoint t[],int ntp)
{
    int k,i,j,p,q,r;
    double s[3],d[3];
    FILE *fp;
    char check;

    if((fp = fopen("coeff.log","w")) == NULL) cprintf("unable to create
coeff.log\n");
    fprintf(fp,"#,tpx,tpy,order,a0,a1,a2,a3,a4,a5,a6\n");

    printf("#,tpx,tpy,x_1,y_1,x_2,y_2,x_3,y_3,ord1,r2_1,ord2,r2_2,ord3,r2_3,be
ta1,beta2,beta3,sumbeta,x,y\n"); /*tpx, tpy coord of triple junctions*/
                                /*x_1,y_1= coordinates of point on first
                                grain segment ; r2_1=correlation coeff of
                                first grain segment*/

    for(k=0;k<ntp;k++)
    {
        if(t[k].g[0].npts<=4||t[k].g[1].npts<=4||t[k].g[2].npts<=4)
        {
            /* printf("\n%f %f has too few points to continue\n",t[k].tx,t[k].ty);*/
            continue;
        }

        if(t[k].tx==-1)
            continue;

```

```
printf("%i,%f,%f",k+1,t[k].tx,t[k].ty);
```

```
for(i=0;i<3;i++)          /* loop to calculate slope for each line*/
```

```
{
```

```
  s[i]=slope(t[k].g[i].a,t[k].g[i].order,t[k].tx);
```

```
/*  printf(" slope %d is %lf ",i,s[i]);*/
```

```
}
```

```
/* printf("\n");*/          /*end of loop for slope*/
```

```
for(i=0;i<3;i++)          /* loop to calculate distance from triple point  
                           to x_1 and y_1*/
```

```
{
```

```
  j=i+1;
```

```
  if(j==3) j=0;
```

```
  get_distance(d,s[i],s[j],t[k],i);
```

```
  t[k].beta[i]=acos((d[0]+d[1]-d[2])/(2*sqrt(d[0])*sqrt(d[1])))*(180/3.1412));
```

```
}
```

```
check = ' ';
```

```
if((t[k].beta[0]+t[k].beta[1]+t[k].beta[2])<360) {
```

```
    check = '*';          /*adding check marks to angles which are not  
                           adding upto 360 degrees in the first run*/
```

```
for(p=0;p<3;p++){
```

```
  q=p+1;
```

```
  if(q==3) q=0;
```

```
  r=q+1;
```

```
  if(r==3) r=0;
```

```
  if(((t[k].beta[p]+t[k].beta[q])>=(t[k].beta[r]-
```

```
0.001))&&((t[k].beta[p]+t[k].beta[q])<=(t[k].beta[r]+0.001)))
```

```
{
```

```
    t[k].beta[p]=180-t[k].beta[p];
```

```
    t[k].beta[q]=180-t[k].beta[q];
```

```
}
```

```
}
```

```
}
```

```
for(p=0;p<3;p++) printf("%i,%f",t[k].g[p].order,t[k].g[p].r2);
```

```
for(p=0;p<3;p++) printf("%f",t[k].beta[p]);
```

```
printf("%f",t[k].beta[0]+t[k].beta[1]+t[k].beta[2]);
```

```

if (check == ' ')
    printf("\n");
else
    printf(",%c\n",check);

for(p=0;p<3;p++)
{
    fprintf(fp,"%i,%f,%f,%i",k+1,t[k].tx,t[k].ty,t[k].g[p].order);
    for(i=0;i<=t[k].g[p].order;i++) fprintf(fp,"%f",t[k].g[p].a[i]);
    fprintf(fp,"\n");
}
}
fclose(fp);
}

```

```

get_distance(double d[], double s1, double s2, triplepoint t, int i)
{
    double c1,c2,y1,y2,x1,x2,m1,m2,k1,k2;
    int j;

    j=i+1;
    if(j==3) j=0;
    c1=t.ty-s1*t.tx; /*equation of straight line*/
    c2=t.ty-s2*t.tx;
    x1=t.g[i].x;
    x2=t.g[j].x;
    y1=t.g[i].y;
    y2=t.g[j].y;

    if (s1 == 0) {
        x1 = t.g[i].x;
        y1 = t.ty;
    }
    else {
        m1=-(1/s1);
        k1=(y1-(m1*x1));
        x1=((k1-c1)/(s1-m1));
        y1=((m1*c1)-(s1*k1))/(m1-s1);
    }
    if (s2 == 0) {

```



```

    x2 = t.g[j].x;
    y2 = t.ty;
}
else {
    m2=-(1/s2);
    k2=(y2-(m2*x2));
    x2=((k2-c2)/(s2-m2));
    y2=((m2*c2)-(s2*k2))/(m2-s2);
}

```

```

d[0]=DISTANCE(t.tx,t.ty,x1,y1);
d[1]=DISTANCE(t.tx,t.ty,x2,y2);
d[2]=DISTANCE(x1,y1,x2,y2);
printf("%lf, %lf",x1,y1);
}

```

```

double slope(float a[], int order, int x)
{
    int i;
    double sum = 0;

    for(i=1;i<=order;i++)
        sum=sum+i*((double) a[i])*pow(x,i-1);

    return(sum);
}

```

```

int tpfit(triplepoint t[], int *ntp)
{
    float
x[MAX_DATA],y[MAX_DATA],sig[MAX_DATA],a[MAX_SIZE];
    int ndat,i,ord,j,l,err,fe,upperord;
    float r2;
/*    FILE *fp;*/

/*    DATA INPUT & CALCULATIONS */
    for (i=0;i<MAX_DATA;i++) sig[i] = 1;

```

```

/*      fp = (FILE *) fopen(fname,"r");
      if (fp == NULL) {
          printf("file error: %i\n",errno);
          exit(0);
      }
*/

(*ntp) = 0;  /*counter for number of triple points*/
do
{
    for(i=0;i<3;i++)
    {
        fe = gb_line(x,y,&ndat);
        err = polyfit(x,y,sig,ndat,1,6,&ord,a,&r2,1,x[0],y[0]);
        if (!err)
        {
            t[*ntp].tx = x[0];
            t[*ntp].ty = y[0];
            t[*ntp].g[i].order = ord;
            t[*ntp].g[i].x=x[ndat-1];
            t[*ntp].g[i].y=y[ndat-1];
            t[*ntp].g[i].r2 = r2;
            t[*ntp].g[i].npts=ndat;
            if ((t[*ntp].g[i].a = (float *)
malloc((ord+1)*sizeof(float))) == NULL)
            {
                cprintf("warning: out of memory in tpfit\n");
                getch();
                return(-1);  /* out of memory */
            }
            for(j=0;j<=ord;j++) t[*ntp].g[i].a[j] = a[j];
        }
        else {
            t[*ntp].tx = -1;
            for (l=i+1;l<3;l++)
                fe = gb_line(x,y,&ndat);
            break;
        }
    }
    (*ntp)++;
} while(fe != -99);

```

```

        return(0);
    }

int gb_line(float x[], float y[],int *ndat)
{
    int num;

    (*ndat) = 0;
    do
    {
        scanf("%i",&num);
        if(num >= 0) {
            if (*ndat < MAX_DATA)
                x[*ndat] = (float) num;
            scanf("%i",&num);
            if (*ndat < MAX_DATA) {
                y[*ndat] = (float) num;
                (*ndat)++;
            }
        }
    } while(num >= 0);
    return(num);
}
[']

```

A.3 Program for curve fitting

```
/*modcf3.c*/
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

#include "cf3.h" /*including header file cf3.h given in appendix A.4*/

/*
main()
{

float x[100], y[100], sig[100], a[MAX_SIZE], xi;
int i, ord;

for (i=0;i<50;i++)
{
xi = ((float) i)/50.0;
x[i] = xi;
y[i] = 0.01+0.02*xi+0.04*xi*xi + ((float) random(100))/100;
sig[i] = 1;
}

polyfit(x,y,sig,50,1,7,&ord,a,0,0,0);

}
*/

int polyfit(float x[],float y[],float sig[],int ndat,int ord1,int ord2,int *ord,float
a[],float *r2,int ianch,float anch_x,float anch_y)
{
/* Polynomial Least Square Fit:
x, y: "ndat" data points; sig: "ndat" standard deviations;
ord1,ord2: search poly order ord1 to ord2 and return best value in *ord (to
implement)
a: on return contains "*ord+1" polynomial coefficients
ianch: non-zero value => set anchor at "anch_x,anch_y"
return value: non-zero => no solution possible */
```

```

int temp,i,j,deg1,deg2,err, ia[MAX_SIZE];
float chisq,chisq1,a1[MAX_SIZE],ftest;
double yc,ym;
long double covar[MAX_SIZE][MAX_SIZE], a_x, st, sr;

void (*fn)(float, long double [], int); /* function pointer */

if (ord2<ord1) SWAP(ord1,ord2)
if ((ord1<0)|| (ord2>MAX_ORDER)) return(-1); /* invalid polynomial
order */
if (ndat<=ord1) return(-1); /* invalid no. of data points */
if (ndat<=ord2) ord2=ndat-1; /* reduce the upper polynomial order */

fn = polyfunc1;
if (ianch) { /* set anchor point */
    anchor_x = anch_x;
    anchor_y = anch_y;
    fn = polyfunc2;
}

for(*ord=ord1;*ord<=ord2;(*ord)++)
{
    for(i=0;i<=*ord;i++) ia[i] = 1;
    if (ianch) { /* set anchor for each order value */
        ia[0] = 0;
        a[0] = anchor_y;
    }
    err = gllsq(x,y,sig,ndat,a,ia,(*ord)+1,covar,&chisq,fn);
    if (err != 0) return(err);/* Error: solution not possible */
    if (ianch) {
        a_x = 1;
        for (i=1;i<=*ord;i++)
        {
            a_x = a_x*((long double) anchor_x);
            a[0] -= a[i]*a_x;
        }
    }
}

if((*ord)>ord1)
{

```

```

    deg1=ndat-(*ord);
    deg2=ndat-((*ord)+1);
    if (chisq1 < chisq) break;
    if((deg2!=0)&&(chisq1!=chisq))
        ftest=(((deg1-deg2)*chisq)/((chisq1-chisq)*deg2));
    else
        break;
    if(ftest > 1)
        break;
}
for(i=0;i<=(*ord);i++)
    a1[i]=a[i];
chisq1=chisq;
}
(*ord)=(*ord)-1;
for(i=0;i<=(*ord);i++) a[i] = a1[i];
ym = 0;          /* calculation of correlation coefficient */
for(i=0;i<ndat;i++) ym += y[i];
ym = ym/ndat;
sr = st = 0;
for(i=0;i<ndat;i++)
{
    st += (long double) ((y[i] - ym)*(y[i] - ym));
    yc = 0;
    for(j=0;j<=(*ord);j++) yc += a[j]*pow((double) x[i],(double) j);
    sr += (long double) ((y[i] - yc)*(y[i] - yc));
}
if((st == 0.0) || (sr == 0))
    (*r2) = 1;
else
    (*r2) = (float) ((st - sr)/st);
/* printf("tp %f %f Ord:%i Deg:%i Chi:%f r2: %f a: ",x[0],y[0],*ord,ndat-
*ord-1,chisq1,*r2);
for(i=0;i<=(*ord);i++) printf("%f ",a1[i]);
printf("\n");
*/

return(0);
}

```

```

void polyfunc1(float x, long double afunc[], int ma)
{
/* returns basis functions (at x) in "afunc" in a polynomial of order "ma-1" */

int i;
afunc[0] = 1;
for (i=1;i<ma;i++)
{
    afunc[i] = ((long double) x) * afunc[i-1];
}
}

```

```

void polyfunc2(float x, long double afunc[], int ma)
{
/* returns basis functions (at x) in "afunc" in a polynomial of order "ma-1"
under constraint that curve is anchored at: (anchor_x, anchor_y) */

int i;
long double x1 = 1, x2 = 1;

afunc[0] = 1;
for (i=1;i<ma;i++)
{
    x1 = x1*x;
    x2 = x2*((long double) anchor_x);
    afunc[i] = x1 - x2;
}
}

```

```

int gllsq(float x[], float y[], float sig[], int ndat, float a[], int ia[], int ma, long
double covar[][MAX_SIZE], float *chisq, void (*gfunc)(float, long double
[], int))
{
/* General Linear Least Square Routine:
x, y: "ndat" data points; sig: "ndat" standard deviations; ma: # of
parameters;
ia: "ma" size with only zero or non-zero values; zero value implies,
corresponding
parameter held fixed to value stored in "a"; gfunc: user routine to supply
"ma"

```

basis functions evalated in array afunc (size: "ma");
 returns ChiSquare in "chisq" and covariance matrix in covar;
 return value: non-zero => no solution possible */

```
int err,i,j,k,l,m,mfit=0;
```

```
float ym,wt,sum,sig2i;
```

```
long double beta[MAX_SIZE],afunc[MAX_SIZE];
```

```
for (j=0;j<ma;j++)
```

```
    if (ia[j]) mfit++;
```

```
if (mfit == 0) return(-1);/* no parameters to be fitted */
```

```
for (j=0;j<mfit;j++)
```

```
{ /* Initialize the (symmetric) matrix */
```

```
    for (k=0;k<mfit;k++) covar[j][k]=0.0;
```

```
    beta[j]=0.0;
```

```
}
```

```
for (i=0;i<ndat;i++)
```

```
{ /* Loop over data to accumulate coeffcients of the normal equations
```

```
*/
```

```
    (*gfunc)(x[i],afunc,ma);
```

```
    ym=y[i];
```

```
    if (mfit < ma)
```

```
{ /* Subtract off dependences on known pieces of the fitting function */
```

```
    for (j=0;j<ma;j++)
```

```
        if (!ia[j]) ym -= a[j]*afunc[j];
```

```
}
```

```
sig2i=1.0/SQR(sig[i]);
```

```
for (j=-1,l=0;l<ma;l++)
```

```
{
```

```
    if (ia[l]) {
```

```
        wt=afunc[l]*sig2i;
```

```
        for (j++,k=-1,m=0;m<=l;m++)
```

```
            if (ia[m]) covar[j][++k] += wt*afunc[m];
```

```
        beta[j] += ym*wt;
```

```
    }
```

```
}
```

```
}
```

```
for (j=1;j<mfit;j++) /* Fill in above the diagonal from symmetry */
```

```
    for (k=0;k<j;k++)
```



```

    covar[k][j]=covar[j][k];
if ((err=gaussj(covar, mfit, beta)) != 0)    /* Matrix solution */
    return(err);    /* no solution possible */
for (j=-1,l=0;l<ma;l++)
    if (ia[l]) a[l]=beta[++j]; /* Partition solution to appropriate coefficients a
*/
*chisq=0.0;    /* and Evaluate ChiSquare of the fit */
for (i=0;i<ndat;i++)
{
    (*gfunc)(x[i],afunc,ma);
    for (sum=0.0,j=0;j<ma;j++)
        sum += a[j]*afunc[j];
    *chisq += SQR((y[i]-sum)/sig[i]);
}
covsrt(covar,ma,ia,mfit);    /* Sort covariance matrix to true order of
fitting coefficients */
return(0);
}

```

```

void covsrt(long double covar[][MAX_SIZE], int ma, int ia[], int mfit)
{
/* Expand in storage the covariance matrix covar, so as to take into account
parameters that are being held fixed (For latter, return zero covariances) */

```

```

    int i,j,k;
    long double temp;

    for (i=mfit;i<ma;i++)
        for (j=0;j<=i;j++) covar[i][j]=covar[j][i]=0.0;
    k=mfit;
    for (j=ma-1;j>=0;j--) {
        if (ia[j]) {
            for (i=0;i<ma;i++) SWAP(covar[i][k],covar[i][j])
            for (i=0;i<ma;i++) SWAP(covar[k][i],covar[j][i])
            k--;
        }
    }
}

```

```

int gaussj(long double a[][MAX_SIZE], int n, long double b[])

```

```

{
/* Gauss-Jordon Method; uses Full Pivoting
a: square matrix of size "n x n"; b: vector of size n
on return: "a" contains inverse of "a" and the solution in "b"
return value: == 0 => no error; != 0 => no solution possible */

int indxc[MAX_SIZE],indxr[MAX_SIZE],ipiv[MAX_SIZE];
int i,icol,irow,j,k,l,ll;
long double big,dum,pivinv,temp,temp1;

for (j=0;j<n;j++) ipiv[j]=0;    /* set pivot tracking elements to zero */

for (i=0;i<n;i++)              /* main loop for pivot search */
{
    big=0.0;
    for (j=0;j<n;j++)
    {
        if (ipiv[j] != 1) /* skip, if diagonal element in jth row is a pivot */
        {
            for (k=0;k<n;k++)
            {
                if (ipiv[k] == 0)
                {
                    if (a[j][k] < 0)
                        temp1 = -a[j][k];
                    else
                        temp1 = a[j][k];
                    if (temp1 >= big)
                    {
                        big = temp1;
                        irow=j;
                        icol=k;
                    }
                }
            }
            else if (ipiv[k] > 1) return(1); /* Error: Singular Matrix-1 */
        }
    }
}
++(ipiv[icol]);    /* irow, icol should contain the position of pivot */

```

```

if (irow != icol)          /* if irow = icol, then pivot already in place */
{
    /* else position pivot by swapping rows */
    for (l=0;l<n;l++) SWAP(a[irow][l],a[icol][l])
    SWAP(b[irow],b[icol])
}
indxr[i]=irow;            /* keep track of the original pivot position */
indxc[i]=icol;

if (a[icol][icol] == 0.0) return(2); /* Error: Singular Matrix-2 */

pivinv=1.0/a[icol][icol];  /* Now divide the pivot row by the pivot
element */
a[icol][icol]=1.0;        /* ??? */
for (l=0;l<n;l++) a[icol][l] *= pivinv;
b[icol] *= pivinv;

for (ll=0;ll<n;ll++)      /* Reduce the rows except the pivot one */
{
    if (ll != icol)
    {
        dum=a[ll][icol];
        a[ll][icol]=0.0;
        for (l=0;l<n;l++) a[ll][l] -= a[icol][l]*dum;
        b[ll] -= b[icol]*dum;
    }
}
/* End of Main Loop */

for (l=n-1;l>=0;l--)      /* Unscramble the solution by interchanging
columns */
{
    if (indxr[l] != indxc[l])
        for (k=0;k<n;k++)
            SWAP(a[k][indxr[l]],a[k][indxc[l]]);
}
return(0);
}
□

```

A.4 Header file called by curve fit program

```
/*cfit.h*/
#define SWAP(a,b) {temp=(a);(a)=(b);(b)=temp;}

static float sqrarg;
#define SQR(a) ((sqrarg=(a)) == 0.0 ? 0.0 : sqrarg*sqrarg)

#define MAX_ORDER 9
#define MAX_SIZE MAX_ORDER+1

static float anchor_x, anchor_y;

int polyfit(float x[],float y[],float sig[],int ndat,int ord1,int ord2,int *ord,float
a[],float *r2,int ianch,float anch_x,float anch_y);
void polyfunc1(float x, long double afunc[], int ma);
void polyfunc2(float x, long double afunc[], int ma);
void covsrt(long double covar[][MAX_SIZE], int ma, int ia[], int mfit);
int gllsq(float x[], float y[], float sig[], int ndat, float a[], int ia[], int ma, long
double covar[][MAX_SIZE], float chisq[], void (*gfunc)(float, long double
[], int));
int gaussj(long double a[][MAX_SIZE],int n, long double b[]);
[]
```

A.5 Program to make comparison of triple points on the two dissector planes

```
/*compare.c*/
#include<stdio.h>
#define MAX 194
typedef struct {long int x,y,dist;
               } typecoord;
typecoord coord[MAX];
int sort(typecoord coord[],int n);
main()
{

    long int a[MAX][2],b[MAX][2],x1,x2,y1,y2; /* a and b array stores the
values of tps from teh two sets */
    FILE *fp;
    int i,j,k,n;

    fp=fopen("tpdata.txt","r");
    i = 0;
    while(fscanf(fp," %ld %ld %ld %ld
",&a[i][0],&a[i][1],&b[i][0],&b[i][1])==4)
    {
        printf(" %ld %ld %ld %ld \n",a[i][0],a[i][1],b[i][0],b[i][1]);
        i++;
    }
    n=i;
    for(i=0;i<n;i++)
    {
        /* calculating the distance for each tp from every tp of the
other set */

        for(j=0;j<n;j++){
            x1 = a[i][0];
            y1 = a[i][1];
            x2 = b[j][0];
            y2 = b[j][1];
            coord[j].dist = (((x2-x1)*(x2-x1))+((y2-y1)*(y2-y1)));
            coord[j].x = x2;
            coord[j].y = y2;
```

```

    }
    sort(coord,n);
    printf(" %ld, %ld, ",x1,y1);
    for(k=0;k<5;k++)
    printf("%ld, %ld, ",coord[k].x,coord[k].y);

    printf("\n");
}
fclose(fp);
}

```

```

int sort(typecoord coord[],int n)
{
    long int i,j,tmp,tmp1,tmp2;
    for (i=0; i<n-1; i++) {
        for (j=0; j<n-1-i; j++)
            if (coord[j+1].dist < coord[j].dist) { /* compare the two neighbors */
                tmp = coord[j].dist;
                tmp1=coord[j].x;
                tmp2=coord[j].y; /* swap a[j] and a[j+1] */
                coord[j].dist = coord[j+1].dist;
                coord[j].x=coord[j+1].x;
                coord[j].y=coord[j+1].y;
                coord[j+1].dist = tmp;
                coord[j+1].x=tmp1;
                coord[j+1].y=tmp2;
            }
    }
}

```

1

242 15 241 15 240 15 239 15 238 15 237 15 236 15 235 15 234 15 233 15 232 15 231 15
 230 15 229 15 228 15 227 15 226 15 225 15 224 15 223 15 222 16 221 17 220 18 219 19
 218 19 217 19 216 19 215 19 214 20 213 20 212 20 211 21 210 21 209 22 208 22 207 22
 206 22 205 22 204 22 203 23 202 23 201 24 200 24 199 24 198 24 197 24 196 24 195 24
 194 23 193 23 192 23 191 22 190 22 189 22 -1
 301 38 302 37 302 36 303 35 303 34 304 33 305 32 306 32 307 31 308 31 309 31 310 31
 311 31 312 31 313 30 314 30 -1
 301 38 302 39 302 40 303 41 303 42 304 43 305 44 306 45 307 46 307 47 307 48 307 49
 308 50 309 51 310 52 310 53 311 54 311 55 311 56 312 57 312 58 313 59 313 60 314 61
 314 62 315 63 315 64 316 65 317 66 318 67 319 68 320 69 -1
 520 43 519 43 -1
 520 43 520 42 -1
 520 43 521 44 522 45 523 46 523 47 524 48 525 49 526 50 527 51 528 52 528 53 529 54
 529 55 530 56 531 57 532 58 533 58 534 59 535 60 536 61 537 62 538 63 539 64 -1
 401 47 401 46 -1
 401 47 402 47 -1
 401 47 400 48 399 48 398 48 397 49 396 49 395 50 394 50 393 50 392 51 391 52 390 53
 389 54 388 55 387 56 386 56 385 57 384 58 383 59 382 60 381 60 380 61 379 61 378 62
 377 63 376 64 375 65 374 66 373 67 372 68 371 69 370 69 369 70 368 71 367 72 366 73
 365 74 364 75 363 75 362 76 -1
 50 62 49 62 -1
 50 62 50 61 -1
 50 62 51 63 52 64 53 65 54 65 55 66 56 67 57 68 58 69 59 70 60 71 61 72 62 73 63 74 64
 75 65 76 65 77 66 78 66 79 67 80 67 81 68 82 69 82 70 83 71 83 72 84 73 85 74 86 75 87
 76 87 77 87 78 88 79 89 80 90 81 91 82 91 83 92 84 92 85 93 86 93 87 93 88 93 89 94 90
 94 91 94 92 94 93 94 94 94 95 95 96 95 97 95 98 95 99 96 100 96 101 96 102 97 103 98
 -1
 14 63 14 62 -1
 14 63 15 63 -1
 14 63 14 64 -1
 539 64 538 63 537 62 536 61 535 60 534 59 533 58 532 58 531 57 530 56 529 55 529 54
 528 53 528 52 527 51 526 50 525 49 524 48 523 47 523 46 522 45 521 44 520 43 -1
 539 64 540 64 -1
 539 64 539 65 -1
 325 66 325 65 -1
 325 66 326 66 -1
 325 66 324 67 323 67 322 68 321 68 320 69 -1
 320 69 321 68 322 68 323 67 324 67 325 66 -1
 320 69 319 68 318 67 317 66 316 65 315 64 315 63 314 62 314 61 313 60 313 59 312 58
 312 57 311 56 311 55 311 54 310 53 310 52 309 51 308 50 307 49 307 48 307 47 307 46
 306 45 305 44 304 43 303 42 303 41 302 40 302 39 301 38 -1
 320 69 321 70 321 71 321 72 321 73 321 74 321 75 321 76 321 77 321 78 321 79 321 80
 321 81 321 82 321 83 321 84 322 85 322 86 322 87 322 88 322 89 323 90 323 91 323 92
 323 93 323 94 323 95 323 96 323 97 323 98 323 99 323 100 323 101 323 102 323 103
 323 104 323 105 323 106 323 107 323 108 323 109 323 110 323 111 323 112 323 113
 323 114 323 115 323 116 323 117 324 118 324 119 324 120 324 121 324 122 324 123
 324 124 324 125 324 126 323 127 323 128 323 129 323 130 323 131 323 132 324 133
 324 134 324 135 324 136 325 137 325 138 326 139 -1

141 72 140 71 140 70 140 69 140 68 140 67 140 66 140 65 140 64 140 63 141 62 141 61
141 60 142 59 142 58 142 57 142 56 142 55 142 54 142 53 142 52 142 51 142 50 142 49
143 48 143 47 143 46 143 45 143 44 143 43 143 42 143 41 143 40 143 39 143 38 143 37
142 36 142 35 141 34 141 33 140 32 140 31 139 30 139 29 138 28 137 27 137 26 136 25
136 24 135 23 135 22 134 21 134 20 133 19 132 18 132 17 131 16 131 15 -1
141 72 142 72 143 72 144 72 145 72 146 72 147 72 148 71 149 70 149 69 150 68 151 67
152 66 153 65 154 65 155 64 155 63 156 62 157 61 158 60 159 59 160 58 161 57 162 56
163 55 164 54 164 53 165 52 166 51 166 50 167 49 168 48 169 47 170 46 171 45 172 44
172 43 173 42 173 41 174 40 175 39 176 38 177 37 178 36 179 35 179 34 180 33 181 32
182 31 183 31 184 30 185 29 185 28 186 27 186 26 187 25 187 24 188 23 189 22 77 -99

B.3 Evaluation of PDA

The data set of appendix B.2 is subjected to algorithm in appendix A.2. this out put is of section S_1 of the dissector, similar output of section S_2 is obtained. The parameters evaluated are highlighted below:

= triple point number

tpx, tpy = triple point coordinate

$x_1, y_1, x_2, y_2, x_3, y_3$ = arbitrary point on vectors \hat{r}_1, \hat{r}_2 and \hat{r}_3 respectively

ord1, r2_1, ord2, r2_2, ord3, r2_3 = order and correlation coefficient of grain boundary segments

beta1, beta2, beta3 = three plane dihedral angles

sumbeta = sum of plane dihedral angle at that triple point.

#, tpx, tpy, $x_1, y_1, x_2, y_2, x_3, y_3$, ord1, r2_1, ord2, r2_2, ord3, r2_3, beta1, beta2, beta3, sumbeta, x, y
4, 189.000000, 22.000000, 185.038903, 14.978055, 288.200453, -6.520143, 150.091494,
78.284764, 1, 0.958875, 2, 0.870281, 5, 0.998546, 103.400429, 140.712845, 115.931717, 360.0
44991
6, 301.000000, 38.000000, 195.128365, 56.698689, 306.141395, 26.489728, 319.239370,
69.441249, 4, 0.968997, 3, 0.977190, 1, 0.987728, 124.100906, 125.827988, 110.116104, 360.0
44998
13, 320.000000, 69.000000, 325.039516, 66.067918, 292.341493, 52.623289, 354.090394,
120.351985, 1, 0.930579, 3, 0.993984, 3, 0.781635, 119.193329, 154.227707, 86.623962, 360.0
44998
14, 141.000000, 72.000000, 154.185542, 20.993644, 174.423261, 80.349695, 108.831854,
103.011990, 3, 0.119484, 5, 0.998520, 3, 0.987638, 89.543457, 122.037247, 148.464294, 360.0
44998
16, 221.000000, 82.000000, 218.661283, 78.650566, 257.380944, 101.782187, 216.267892,
88.411804, 2, 0.975546, 2, 0.993865, 3, 0.962158, 153.478760, 97.905502, 108.660736, 360.04
4998
22, 545.000000, 105.000000, 540.836580, 64.809744, 517.500365,
131.719387, 587.473649,
119.991877, 2, 0.950238, 2, 0.991856, 2, 0.991766, 128.277237, 116.397568, 115.370201, 360.
045006

26,510.000000,124.000000,547.339081, 112.793877,464.842053,
 106.872816,512.575528,
 139.240721,3,0.991924,4,0.963480,2,0.916537,142.542007,120.377213,97.125786,360.0
 45006
 28,326.000000,139.000000,337.384008, 71.951594,361.829520,
 137.608939,311.333820,
 176.741123,3,0.774139,3,0.830942,3,0.985906,78.150177,113.473640,168.421188,360.0
 45006
 29,514.000000,139.000000,507.475944, 125.192630,535.783195,
 140.348496,491.713823,
 189.675221,2,0.926352,2,0.989280,3,0.980613,118.848228,110.210533,130.986237,360.
 044998
 30,370.000000,141.000000,362.096191, 142.505125,382.375173,
 114.761999,410.493264,
 238.775649,3,0.790541,3,0.983100,4,0.999042,126.048462,132.268951,101.727585,360.
 044998
 31,362.000000,142.000000,367.107203, 137.623906,335.005233,
 124.836534,355.933006,
 184.175903,3,0.817308,3,0.826396,3,0.793411,106.973343,114.276947,138.794708,360.
 044998
 36,102.000000,190.000000,93.225074, 99.952545,153.037462, 211.347064,33.434176,
 254.958492,1,-
 1.163141,3,0.973940,4,0.997891,118.278297,113.864037,127.902672,360.045006
 38,529.000000,201.000000,531.222104, 153.822059,519.249695,
 206.299329,591.638038,
 213.392052,3,0.761801,2,0.933288,2,0.988666,121.236076,140.302628,98.506302,360.0
 45006
 40,149.000000,221.000000,137.552148, 180.058976,246.574116,
 228.019208,129.531425,
 246.601946,5,0.989639,5,0.963545,3,0.992455,109.750282,123.151352,127.143364,360.
 044998
 41,451.000000,222.000000,499.948491, 199.380530,353 224641,
 181.682388,454.213634,
 261.390491,3,0.983708,6,0.999056,3,0.849216,132.805801,117.087494,110.151695,360.
 044991
 42,257.000000,235.000000,247.019468, 236.166556,310.504040,
 176.628778,341.079454,
 323.818081,1,0.728568,1,0.991068,3,0.987218,139.192947,94.072784,126.779266,360 0
 44998
 45,584.000000,252.000000,552.137579, 189.251251,631.073037,
 252.232820,559.269855,
 284.128840,3,0.994823,3,0.990283,3,0.994494,117.218475,127.318657,115.507866,360.
 044998
 46,459.000000,261.000000,435.429845, 245.141296,526.803188,
 243.923029,447.639138,
 285.600672,2,0.950700,3,0.977888,2,0.894718,131.946106,128.940674,99.158226,360.0
 45006
 51,563.000000,287.000000,500.031294, 292.211129,579.987160,
 250.150148,569.223803,

300.713774,5,0.994832,2,0.994269,3,0.983531,119.494713,130.857193,109.693092,360.044998
 52,455.000000,289.000000,457.056193, 260.857976,346.016923,
 330.016189,500.991958,
 316.013692,2,0.713324,3,0.994060,1,0.984857,114.817245,128.963959,116.263786,360.044991
 55,573.000000,299.000000,563.127431, 286.896061,632.199136,
 288.936322,559.848006,
 338.292184,3,0.987681,2,0.991606,3,0.957694,119.569443,118.169228,122.306335,360.045006
 57,78.000000,320.000000,53.404932, 288.746559,44.177936, 346.382090,59.670518,
 337.303324,2,0.993361,4,0.771361,6,0.966311,90.234795,174.604034,95.161171,360.000000,*
 58,343.000000,322.000000,249.349618, 244.210223,457.150689,
 300.328216,338.897731, 367.721656,3,0.989624,4,0.993320,1,-
 2.085013,129.552002,105.890045,124.602959,360.045006
 61,25.000000,333.000000,42.832672, 321.524222,46.136218, 298.301244,32.788625,
 355.025423,3,0.961198,5,0.843150,6,0.995228,154.106125,129.194504,76.699379,360.000008,*
 64,107.000000,347.000000,10.882196, 375.476277,137.577195,
 271.559529,127.061884,
 460.752726,2,0.991375,3,0.970328,2,0.985559,128.582230,147.952957,83.509819,360.045006
 65,47.000000,350.000000,25.928435, 351.081940,47.796537, 328.028878,47.740974,
 350.729626,5,0.996002,2,0.099828,3,0.943679,95.027489,132.498184,132.519333,360.045006
 68,370.000000,410.000000,350.899564, 365.205236,458.072684,
 380.254347,361.370137,
 422.128805,3,0.995838,1,0.964663,2,0.936394,94.443275,144.112427,121.489304,360.045006
 70,219.000000,457.000000,170.611496, 479.470269,278.635686,
 337.799344,238.446494,
 474.503956,3,0.968878,4,0.992115,1,0.969967,141.505173,105.425179,113.114647,360.044998
 71,137.000000,459.000000,70.727878, 375.941369,158.713392,
 450.950264,123.726194,
 485.806478,4,0.986472,4,0.984992,4,0.972689,108.258652,136.701477,115.084862,360.044991
 73,448.000000,465.000000,409.984019, 465.103027,470.013531,
 462.180457,453.342319,
 479.505634,3,0.967119,3,0.864450,1,0.627191,172.878052,77.090088,110.076866,360.045006

B.4 Comparison of Triple Points of Disector Sections

TP x1	TP TRIPLE POINTS OF SECOND SECTION y1 OF THE DISECTOR										
189	22	202	32	260	26	147	84	332	93	369	94
301	38	260	26	332	93	369	94	202	32	334	156
320	69	332	93	369	94	260	26	334	156	395	134
141	72	147	84	202	32	260	26	135	260	332	93
221	82	202	32	260	26	147	84	332	93	334	156
545	105	519	151	521	159	395	134	369	94	586	278
510	124	519	151	521	159	395	134	369	94	586	278
326	139	334	156	332	93	369	94	395	134	255	245
514	139	519	151	521	159	395	134	369	94	586	278
370	141	395	134	334	156	369	94	332	93	519	151
362	142	334	156	395	134	369	94	332	93	255	245
102	190	135	260	113	274	50	265	158	288	147	84
529	201	521	159	519	151	586	278	633	300	395	134
149	221	135	260	113	274	158	288	126	317	50	265
451	222	521	159	519	151	395	134	334	156	586	278
257	235	255	245	334	156	158	288	135	260	337	335
584	252	586	278	633	300	521	159	519	151	491	370
459	261	491	370	521	159	519	151	586	278	395	134
563	287	586	278	633	300	491	370	521	159	519	151
455	289	491	370	337	335	586	278	521	159	340	380
573	299	586	278	633	300	491	370	521	159	519	151
78	320	126	317	113	274	50	265	135	260	158	288
343	322	337	335	340	380	370	427	255	245	491	370
25	333	50	265	126	317	113	274	135	260	158	288
107	347	126	317	113	274	158	288	135	260	50	265
47	350	50	265	126	317	113	274	135	260	158	288
370	410	370	427	340	380	337	335	491	370	222	459
219	457	222	459	136	458	340	380	370	427	126	317
137	459	136	458	222	459	126	317	158	288	113	274
448	465	370	427	491	370	340	380	337	335	600	600